

John von Neumann-Institut für Computing (NIC)

Stefan Birmanns

**Haptisches Rendern zum Einpassen von
hochaufgelösten Molekülstrukturdaten in
niedrigaufgelöste Elektronenmikroskopie-
Dichteverteilungen**

NIC-Serie Band 18

ISBN 3-00-010004-0

Zentralinstitut für Angewandte Mathematik

Die Deutsche Bibliothek - CIP-Einheitsaufnahme
Ein Titeldatensatz für diese Publikation ist bei
Der Deutschen Bibliothek erhältlich.

Herausgeber: NIC-Direktorium
Vertrieb: NIC-Sekretariat
Forschungszentrum Jülich
52425 Jülich
Deutschland
Internet: www.fz-juelich.de/nic
Druck: Graphische Betriebe, Forschungszentrum Jülich

© 2003 John von Neumann-Institut für Computing

Es ist erlaubt, dieses Werk oder Teile davon digital oder auf Papier zum persönlichen Gebrauch oder zu Lehrzwecken zu vervielfältigen, vorausgesetzt die Kopien werden nicht kommerziell genutzt. Kopien müssen diese Copyright-Notiz und das volle Zitat auf ihrer Titelseite enthalten. Andere Vervielfältigung bedarf der vorherigen schriftlichen Genehmigung des oben genannten Herausgebers.

Haptisches Rendern zum Einpassen von hochaufgelösten Molekülstrukturdaten in niedrigaufgelöste Elektronenmikroskopie-Dichteverteilungen

Von der Fakultät für Elektrotechnik und Informationstechnik der Rheinisch-Westfälischen
Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors
der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Informatiker

Stefan Birmanns

aus Aachen

Berichter: Universitätsprofessor Dr. rer. nat. Friedel Hoßfeld
 Universitätsprofessor Christian H. Bischof, Ph.D.

Tag der mündlichen Prüfung: 23.1.2003

Haptisches Rendern zum Einpassen von hochaufgelösten Molekülstrukturdaten in niedrigaufgelöste Elektronenmikroskopie-Dichteverteilungen

Das Forschungsgebiet der strukturellen Biologie hat in den letzten Jahren große Erfolge in der Strukturermittlung von immer größeren biologischen Einheiten erzielen können. Mittlerweile sind auch große Makromoleküle, sogenannte zelluläre Maschinen, in den Mittelpunkt des Interesses gerückt. Um ihre Arbeitsweise verstehen zu können, müssen Fragen nach strukturellen und funktionalen Zusammenhängen auf atomarer Ebene beantwortet werden.

Verschiedene Strukturermittlungs-Techniken wurden in den letzten Jahren entwickelt, um solche Mehrkomponenten-Moleküle untersuchen zu können. Besonders erfolgreiche Techniken in der Elektronenmikroskopie, die wesentliche Arbeitsschritte der Moleküle aufnehmen können, sind jedoch in der Auflösung stark begrenzt. Da sich eine zelluläre Maschine aus meist bekannten Teilkomponenten zusammensetzt, kann ein Gesamtmodell auch durch die Kombination von verschiedenen aufgelösten Abbildungen zusammengesetzt werden. Dieses Verfahren hat sich zu einem wichtigen Werkzeug in der strukturellen Biologie entwickelt. Sowohl interaktive als auch algorithmische Methoden haben sich etabliert.

Diese Dissertationsschrift schlägt eine neue interaktive Methode zur Kombination von verschiedenen aufgelösten Daten vor. Der Benutzer bewegt interaktiv eine hochaufgelöste Teilkomponente in die niedrigaufgelöste Dichteverteilung, wobei er von haptischem Rendern, also einem künstlich hervorgerufenen taktilen Sinnesindruck, unterstützt wird. In dieser Methode wird die Hand des Anwenders durch das haptische Rendern in die Richtung einer besseren Abbildungsposition gezogen, dabei bildet die physikalisch basierte Modellierung eines Kraftfeldes die Grundlage der Berechnung. Erprobte Bewertungsfunktionen, die auch von algorithmischen Programmen eingesetzt werden, entsprechen der Potentialfunktion des Feldes. Da der Einsatz von haptischem Rendern eine hohe Kraftaktualisierungsfrequenz von zirka 1000 Hz erfordert, wird die Methode der Vektorquantisierung zur Reduktion der Komplexität der Berechnungen eingesetzt. Durch die Vektorquantisierung können Datensätze auf wenige Merkmalsvektoren reduziert werden, wobei in dieser Arbeit sogenannte topologieerhaltende neuronale Netzwerke verwendet werden. Durch einen iterativen Prozess wird ein künstliches neuronales Netzwerk so trainiert, dass die Gewichtsvektoren der Neuronen ein Signal möglichst gut beschreiben.

Im Gegensatz zu existierenden interaktiven Ansätzen wird der Benutzer darüber hinausgehend durch Virtual-Reality-Techniken unterstützt. Gerade die Orientierung im virtuellen Raum, die für das Zusammensetzen der dreidimensionalen Strukturen notwendig ist, kann so erleichtert werden.

Diese Methode wurde zum Nachweis der Praxistauglichkeit in Form des Programmes "SenSitus" implementiert. Es unterstützt Shared-Memory-Parallelrechnersysteme, um sowohl die Kraftberechnung für das haptische Rendern als auch die graphische Darstellung zu beschleunigen. Letztere kann sowohl auf Standard-Workstations (diverse Unix-Derivate, PCs unter Linux und Windows) als auch Virtual-Reality-Systemen mit Mehrseitenprojektion ausgegeben werden. Das Programm wurde an verschiedenen Anwendungsbeispielen getestet, und die Geschwindigkeit und Präzision des haptischen Renderns wurde systematisch untersucht.

Haptic rendering for the fitting of high-resolution molecular structures into low-resolution density maps from electron microscopy

The research area of structural biology enjoyed much success in recent years in solving the structure of large-scale molecules. Very large macromolecular assemblies, the so-called "cellular machines" became the subject of considerable research efforts. To understand their functioning, questions about structural and functional relationships on atomic level have to be answered.

Various structure solving techniques were developed in recent years to analyze these multicomponent-molecules. The particularly successful methods in electron microscopy, which are able to acquire different working steps of the molecules, are limited in the attainable resolution. Since cellular machines often consist of components whose structure is known at atomic detail, a complete model can be assembled with the help of low-resolution micrographs. Both interactive and algorithmic methods were established that have become important tools for concurrent structural biology research.

This thesis suggests a new interactive method for the combination of multi-resolution data. Supported by haptic rendering, the user moves a high-resolution component into the low-resolution density map interactively. Haptic rendering creates an artificial tactile sensation. In this thesis the haptic rendering is used to guide the user's hand towards a better fitting location, whereby a physical force field forms the basis of the calculation. A well-established linear correlation, typically used in algorithmic methods, acts as a potential energy function. Since the use of haptic rendering requires force-updates of about 1000 Hz, vector quantization is used to reduce the complexity of the calculations. Vector quantization is able to describe data sets by a small number of so-called codebook vectors that are computed here with topology-representing networks. These artificial neural networks are trained in an iterative process so that the neural vectors describe the signal as close as possible.

In contrast to existing interactive graphics software, the user is supported by virtual-reality techniques, such as immersive stereo displays and user-centered projection. This facilitates the navigation in the virtual space that is crucial for the fitting of the three-dimensional structures.

For validation and proof of concept the method was implemented in the novel program "SenSitus". The software supports shared-memory parallel computers to speed up haptic rendering and visualization. It can be used on standard-workstations (several Unix-derivatives, PCs with Linux and Windows), but also on multi-display virtual-reality systems. The program was tested on several structural application examples and the speed and accuracy of the haptic rendering was systematically analyzed.

Danksagung

Die vorliegende Dissertation wurde am Zentralinstitut für Angewandte Mathematik (ZAM) am Forschungszentrum Jülich geschrieben. Mein besonderer Dank gilt Herrn Prof. Dr. Hoßfeld, der während der Anfertigung der Arbeit Direktor des ZAM und Inhaber des Lehrstuhls für Technische Informatik und Computerwissenschaften an der Rheinisch-Westfälischen Technischen Hochschule (RWTH) Aachen war. Sein großes Interesse und die hervorragende Betreuung haben diese Arbeit erst möglich gemacht. Auch danke ich Herrn Prof. Bischof, Ph.D., für die Übernahme des Koreferates und seine Unterstützung dieser Arbeit. Ebenfalls danke ich Herrn Dr. Rüdiger Esser, Leiter der Abteilung Programmiertechniken am ZAM.

Bedanken möchte ich mich bei Herrn Dr. Herwig Zilken, der mich vor allem bei dem Einsatz von Virtual Reality- und Visualisierungstechniken durch viele motivierende Diskussionen sehr unterstützt hat. Ohne ihn wäre die Arbeit in dieser Form sicher nicht zustande gekommen.

Ebenso gilt mein Dank Herrn Prof. Wriggers, Ph.D., der es immer verstanden hat, mir die physikalischen Aspekte der hier behandelten Problematik näher zu bringen. Bedanken möchte ich mich auch für seine Einladung an das Scripps Research Institute nach San Diego. Während dieser Zeit konnten viele Details des Lösungsansatzes durch intensive Diskussionen weiter verbessert werden. Ebenso möchte ich den anderen Mitgliedern der Computational Structural Biology Group für ihre Unterstützung danken. Vor allem Pablo Chaón, Ph.D., Julio Kovacs, Ph.D. und Yao Cong, Ph.D., haben durch Diskussionen über ihre jeweils eigenen Ansätze zur Lösung des Einpassproblems viele Anregungen für meine eigene Arbeit geliefert.

Ebenfalls möchte ich der Virtual-Reality-Abteilung am Rechenzentrum der RWTH Aachen danken. Vor allem Herr Dr. Thorsten Kuhlen, Leiter der VR-Abteilung, hat durch die enge Zusammenarbeit mit dem ZAM und den regen Gedankenaustausch Anteil am Gelingen dieser Arbeit gehabt.

Meinen Kollegen am ZAM möchte ich für ihre Hilfsbereitschaft und die nette Gemeinschaft danken. Vor allem gilt mein Dank Felix Wolf, der zur gleichen Zeit Doktorand am ZAM war. Durch das Teilen der vielen kleinen und größeren Sorgen eines Doktoranden waren diese viel leichter zu bewältigen. Ebenfalls gilt mein Dank Frank Delonge, der mich bei der praktischen Umsetzung durch viele Tipps und Ideen unterstützt hat. Gedankt sei natürlich auch allen anderen Kollegen und Studenten am ZAM, die diese Arbeit durch ihre Diskussionsbereitschaft sehr bereichert haben.

Schliesslich möchte ich meiner Familie und meinen Freunden für die Aufmunterung und Motivation während meiner Zeit als Doktorand danken. Besonders möchte ich meiner Frau Maria für das Korrekturlesen der Arbeit und die liebevolle Unterstützung während der Promotion danken.

Inhaltsverzeichnis

Abbildungsverzeichnis	v
1 Einleitung	1
1.1 Haptisches Rendern	3
1.2 Visualisierung und Virtual Reality	4
1.3 SenSitus	4
2 Visualisierung und Virtual Reality	7
2.1 Visualisierung	7
2.1.1 Visualisierung von Molekülstrukturdaten	9
2.1.2 Visualisierung von Volumendaten	12
2.1.3 Visualisierungssoftware	13
2.2 Virtual Reality	15
2.2.1 VR-Systeme	17
2.2.2 VR-Software	23
3 Haptisches Rendern	25
3.1 Terminologie	25
3.2 Wahrnehmung	26
3.2.1 Tastsinn	26
3.2.2 Propriozeption	27
3.2.3 Regelung der Muskelaktivität	28
3.3 Kraftrückkopplungsgeräte	28

3.3.1	Eigenschaften von Kraftrückkopplungsgeräten	30
3.3.2	Auswahl eines Gerätes	30
3.3.3	SensAble Phantom	31
3.4	Qualität des Haptischen Renderns	33
3.5	Kraftberechnung	35
3.6	Projekte	37
3.6.1	GROPE	38
3.6.2	IMD	39
4	Interaktiver Einpassprozess	41
4.1	Motivation	41
4.2	Abbildungsproblem	44
4.3	Entwicklung einer Bewertungsfunktion	46
4.3.1	Filter in der Bildverarbeitung	47
4.3.2	Erzeugen einer Dichteverteilung	49
4.3.3	Korrelation	51
4.3.4	Interpolation	52
4.4	Einfaches Verfahren	54
4.5	Topology Representing Networks	58
4.5.1	Neuronale Netzwerke	58
4.5.2	Voronoi-Zellen und Delaunay-Triangulierung	61
4.5.3	Hebb-Regel erzeugt Delaunay-Triangulierung	63
4.5.4	Neural Gas Algorithm	64
4.5.5	Delaunay-Triangulierung und Topology Preserving Maps	65
4.5.6	Algorithmus	66
4.6	Erweitertes Verfahren	67
4.6.1	Vektorquantisierung	67
4.6.2	Korrelationskoeffizient	69
4.6.3	Beschreibung des Verfahrens	70
4.7	Kraftberechnung	70

4.7.1	Mechanik eines Teilchens	71
4.7.2	Teilchensysteme	75
4.7.3	Mechanik eines starren Körpers	78
4.7.4	Anwendung der Gesetze der Mechanik zur Kraftberechnung . . .	79
4.8	SensAble Phantom	81
4.8.1	Koordinaten	81
4.8.2	Krafterzeugung	82
4.8.3	Koordinatensysteme	83
4.9	Zusammenfassung	85
5	Verwandte Arbeiten	87
5.1	Interaktive Abbildungsmethoden	87
5.2	Algorithmische Abbildungsmethoden	89
5.3	Vergleich interaktiver und algorithmischer Methoden	94
5.4	Anwendungsgebiete	96
6	SenSitus	97
6.1	L.I.V.E.	98
6.1.1	Aufbau von L.I.V.E.	98
6.1.2	Datenkanäle	99
6.1.3	Treibermodule	100
6.1.4	Netzwerkzugriff	101
6.2	Scientific Visualization Toolkit	103
6.2.1	Aufbau von SVT	103
6.2.2	svt_core	106
6.2.3	SVT Beispielprogramm	115
6.2.4	svt_pdb	116
6.3	Testanwendung	121
6.4	SenSitus	123
6.4.1	Implementierung	123
6.4.2	Benutzerführung	125

7	Beispielanwendungen	131
8	Zusammenfassung und Ausblick	143
	Literaturverzeichnis	147

Abbildungsverzeichnis

2.1	Visualisierung einer Molekülbeschreibung mit SenSitus	8
2.2	CPK Darstellung eines Molekül	11
2.3	CAVE	20
2.4	VR-Workbench oder Holobench	21
3.1	Phantom Desktop	33
3.2	Phantom 6DOF	34
3.3	Szenengraphen und Haptik	36
3.4	GHOST	37
4.1	Mittelwert-Filter	48
4.2	Berechnete Dichteverteilungen	50
4.3	Interpolation 0. Ordnung	53
4.4	Interpolation 1. Ordnung	53
4.5	Initialisierungsphase	56
4.6	Einpassprozess	57
4.7	Natürliches neuronales Netzwerk	59
4.8	Künstliches neuronales Netzwerk	60
4.9	2D-Voronoi-Diagramm und Delaunay-Triangulation	62
4.10	Vektorquantisierung (2REC Monomer)	68
4.11	Codebookvektoren und 2REC Struktur (Tube-Modus)	68
4.12	Vorbereitungsphase	70
4.13	Hauptschleife	71
4.14	Drehmoment	72

4.15	Schwerpunkt	77
4.16	Drehmoment und Translationskraft	83
4.17	Koordinatensysteme bei Abbildungsprozess	84
5.1	Das interaktive Einpassprogramm „O“	89
5.2	Programmablauf einer algorithmischen Lösung	90
5.3	Übersicht über das Situs Programmpaket	93
6.1	Übersicht über die Architektur von SenSitus	97
6.2	Datenkanäle am Beispiel des Phantom	99
6.3	L.I.V.E.-Treibermodule	100
6.4	L.I.V.E.-Netzwerkzugriff	102
6.5	Übersicht über SVT	104
6.6	Sichtpyramide und Frustum	106
6.7	Projektion am Beispiel der Holobench	107
6.8	Hierarchische Beschreibung einer Szene	108
6.9	Ein Würfel des Marching-Cube-Algorithmus	110
6.10	Oberflächendarstellung eines Volumens	111
6.11	Volumerendering mit 3D-Texturen	112
6.12	Volumerendering mit Volslice3D	114
6.13	Space-Partitioning zur Berechnung der Atombindungen	116
6.14	Tube-Modus	120
6.15	Testanwendung	121
6.16	Lennard-Jones-Potential (Argon)	122
6.17	Aufbau von SenSitus	124
6.18	SenSitus	125
6.19	SenSitus: Hauptfenster	125
6.20	SenSitus: Transformationsdialog	126
6.21	SenSitus: PDB-Einstellungsdialog	127
6.22	SenSitus: Einstellungsdialog Oberflächendarstellung	128
7.1	Kraftausgabe bei einem einzelnen Atom	132

7.2	10, 20, 40 und 100 Codebookvektoren - 1ATN	133
7.3	Kraftausgabe bei Verschiebung aus perfekter Abbildungsposition (1ATN)	134
7.4	Molekül 1CLL und Dichteverteilungen (10Å, 20Å, 30Å Auflösung) . . .	135
7.5	Kraftausgabe bei Verschiebung aus perfekter Abbildungsposition (1CLL) .	136
7.6	Kraftausgabe bei Rotation aus perfekter Abbildungsposition (1CLL) . . .	136
7.7	Kraftausgabe bei unterschiedlicher Auflösung der Dichteverteilung (1CLL)	138
7.8	Geschwindigkeit der Kraftausgabe auf einem Single-CPU System	139
7.9	Geschwindigkeit der Kraftausgabe auf einem Dual-CPU System	139

Kapitel 1

Einleitung

In den letzten Jahrzehnten hat das Forschungsgebiet der strukturellen Biologie immer mehr an Bedeutung gewonnen. Die Ermittlung und Untersuchung der Struktur von biologischen Molekülen ist für die Analyse von zellulären Prozessen von großer Bedeutung. Durch das Wissen über den atomaren Aufbau und die genaue Funktion bestimmter Moleküle in einer Zelle können Medikamente entwickelt werden, die in Prozesse korrigierend eingreifen.

Das Wissen über die Arbeitsweise biologischer, zellulärer Strukturen hat sich in der letzten Zeit deutlich verändert. War vor einigen Jahrzehnten ein eher undeutliches Bild über die dort ablaufenden Prozesse vorherrschend, so wird heute davon ausgegangen, dass sogenannte “zelluläre Maschinen” [2] eine wichtige Rolle spielen. Diese Maschinen sind große Makromoleküle, die — genau wie ihre makroskopischen Namensvettern — komplexe Zustandsänderungen vollführen können. Diese Änderungen verursachen Reaktionen, deren Produkte zum Beispiel von anderen Maschinen weiterverarbeitet werden.

Eine genauere Untersuchung solcher zellulärer Maschinen kann mit Hilfe von verschiedenen Mikroskopietechniken erfolgen. Ein ideales Mikroskop würde es erlauben, eine zelluläre Maschine bei ihrer Arbeit, eingebettet in ihr natürliches Umfeld, in atomarer Auflösung zu beobachten. Leider ist ein solches Mikroskop mit den derzeitigen technischen Möglichkeiten nicht realisierbar.

Es existieren hochaufgelöste Elektronen- und Diffraktionstechniken, welche in der Lage sind, biologische Moleküle mit nahezu atomarer Auflösung zu betrachten. Sie benötigen jedoch eine sehr aufwendige Präparation des zu untersuchenden Objektes. Hierbei wird dieses sowohl aus der Zelle gelöst, als auch in seinem biologischen Zustand verändert. Eine Untersuchung des Moleküls in seiner natürlichen Umgebung ist so genauso wenig möglich, wie die Analyse von allen biologischen Konfigurationen. Ebenso besitzen diese Techniken in der Regel Einschränkungen bei der Untersuchung von großen Strukturen.

Neben den sehr hochaufgelösten Strukturanalysen existieren auch niedriger aufgelöste Methoden auf der Grundlage der Elektronenmikroskopie. Diese sind in der Lage, die biologischen Strukturen mit weniger Präparationsaufwand und in anderen biophysikalischen

Zuständen abzubilden. Sie sind für die Beobachtung von zellulären Maschinen also eigentlich besser geeignet, jedoch besitzen ihre Aufnahmen eine zu geringe Auflösung, um exakte Aussagen über das Molekül zu erlauben.

Aus dieser Problemstellung heraus haben sich Methoden etabliert, die die Stärken der verschiedenen Techniken ausnutzen, indem sie die Ergebnisse miteinander kombinieren. Gerade die Kombination von niedrigaufgelösten, aber großen Datensätzen, mit hochaufgelösten, kleinen Datensätzen wird oft angewendet. Mit Hilfe von sogenannten Multiskalenverfahren können so komplexe Makromoleküle, wie die erwähnten zellulären Maschinen, untersucht werden.

Ein häufig eingesetztes Verfahren ist das sogenannte “Rigid-Body Docking”. Hierbei werden die hochaufgelösten Strukturdaten als starrer Körper modelliert und in die niedrigaufgelösten Datensätze eingepasst. Die Strukturdaten werden dabei zumeist mit Hilfe der Röntgenkristallographie ermittelt, die niedrigaufgelösten Dichteverteilungen mit Elektronenmikroskopen.

Das Problem bei diesem Ansatz liegt in dem Prozess des Einpassens, im folgenden Einpassprozess genannt. Dieser Vorgang ist nicht trivial und nur schwer mit Hilfe eines vollautomatischen Algorithmus zu lösen. Ein Hindernis stellt die Komplexität des Problems dar. Wie im Verlauf der Arbeit noch weiter ausgeführt wird, ist bei vielen Lösungen die benötigte Rechenzeit sehr hoch. Ebenfalls laufen viele Algorithmen Gefahr, statt des globalen Maximums nur ein lokales Maximum zu finden.

Daneben ist die Beschaffenheit der niedrigaufgelösten Daten ein Problem. Die üblicherweise verwendeten Mikroskopietechniken erzeugen Datensätze, deren Kontrast schwach ist und die durch Rauschen gestört sind. Ausserdem ist die Auflösung meist sehr gering, so dass für die automatischen Algorithmen der in den Daten enthaltene Informationsgehalt nicht ausreichend ist. Viele Algorithmen berechnen unter solchen Bedingungen falsche positive Ergebnisse, erkennen also gute Abbildungspositionen an Stellen, an denen eine solche nicht vorliegt.

In der Vergangenheit wurden unterschiedliche Verfahren [68, 60, 59, 45, 46] entwickelt, welche sich für bestimmte Anwendungsgebiete eignen. Einen universellen Algorithmus, der das Abbildungsproblem in allen Fällen präzise löst, existiert zur Zeit nicht. Im Gegenteil, trotz aller Erfolge bei der Suche nach neuen algorithmischen Ansätzen zur Lösung des Problems, werden zur Zeit immer noch in vielen Fällen interaktive Programme bevorzugt.

Der Experte, der eventuell sogar selbst die Daten ermittelt hat, verfügt über ein großes Kontextwissen und kennt die Schwächen der Messverfahren genau. Er kann durch sein Wissen und seinen Überblick über die Daten leichter globale Lösungen erkennen. Eine interaktive Lösung kann diese Erfahrung ausnutzen und die Einschränkungen der Algorithmen vermeiden. Es wird dabei auf zwei- oder dreidimensionale Visualisierungsprogramme zurückgegriffen, mit deren optischen Informationen der eigentliche Einpassprozess durchgeführt werden kann. Die gesamte Lösung des Problems wird bei einem solchen Ansatz

jedoch auf den Experten übertragen, der aufgrund der Komplexität der Aufgabe hier ebenfalls auf Schwierigkeiten stösst.

In der vorliegenden Arbeit wird ein neuer Lösungsvorschlag für das gerade geschilderte Abbildungsproblem vorgestellt. Es wird versucht, die Stärken der interaktiven Lösungen mit denen der Algorithmen zu verbinden. Hierzu wird ein interaktives Programm realisiert, welches zusätzliche Informationen per haptischem Rendern an den Benutzer weitergibt. Diese Informationen entsprechen den Kriterien, die auch ein Algorithmus benutzt, um eine geeignete Abbildungsposition zu suchen. Dies gibt dem Benutzer die Möglichkeit während des Abbildungsvorgangs über die Einschätzung des Computers informiert zu werden, ohne dieser jedoch folgen zu müssen. So wird die Komplexität des Vorgangs für den Experten reduziert, und trotzdem kann sein Kontextwissen genutzt werden.

Neben dem eigenen Ansatz untersucht die Arbeit bestehende Lösungen und beschreibt sowohl die interaktiven als auch die algorithmischen Programme.

1.1 Haptisches Rendern

Der Begriff “multimodale Interaktionsmechanismen” beschreibt Methoden, bei denen mit einem Computer nicht ausschliesslich über die von ihm bereitgestellten visuellen Informationen interagiert wird. Neben dem Auge stehen dem Menschen noch weitere Sinnesorgane zur Verfügung, die von einem Computerprogramm angesprochen werden können.

Ein bei der Interaktion mit unserer natürlichen Umwelt sehr wichtiger Sinn ist der Tastsinn. Der Begriff haptisches Rendern beschreibt die künstliche Erzeugung eines solchen Sinnesindrucks, wobei hier der Begriff Tastsinn recht weit gefasst wird. Sowohl die Wahrnehmung, welche beim Abtasten einer Oberfläche entsteht, als auch ein reiner Krafteindruck werden hierunter verstanden. Die vorliegende Arbeit untersucht, wie diese Sinneseindrücke beim Menschen entstehen, und wie sie von einem Computer ausgelöst werden können.

Zum haptischen Rendern werden geeignete Geräte benötigt, die in der Lage sind, diesen Sinneseindruck beim Benutzer hervorzurufen. Die vorliegende Arbeit gibt sowohl einen Überblick über die zur Zeit erhältlichen Geräte, als auch über die verschiedenen Entwicklungen in diesem Forschungsgebiet in den letzten Jahren.

Weiterhin wird beschrieben, wie bei der Realisierung eines Programms vorgegangen wird, welches haptisches Rendern einsetzen soll. Die Berechnung einer zur Ausgabe an den Benutzer geeigneten Kraft ist in der Regel aufwendig, und gleichzeitig erfordert die hohe Leistungsfähigkeit des menschlichen Tastsinnes eine sehr hohe Neuberechnungsfrequenz. Dies führt zu Problemen bei der Realisierung von solchen Programmen, auf die im Rahmen der Arbeit noch genauer eingegangen wird.

1.2 Visualisierung und Virtual Reality

Bei einem interaktiven Lösungsansatz kommt der optischen Darstellung eine wichtige Rolle zu. Es müssen im Rahmen der vorliegenden Problemstellung die Strukturdaten visualisiert werden. Diese liegen heute zumeist als PDB (Protein Data Bank) Dateien [9] vor, in denen die Atome eines Moleküls und vor allem ihre Positionen im Raum genauer beschrieben werden.

Die Visualisierung dieser Dateien sollte verschiedene Kriterien erfüllen. Durch die große Anzahl von Atomen ergeben sich Probleme sowohl hinsichtlich der Übersichtlichkeit als auch hinsichtlich der Geschwindigkeit der Darstellung. Da neben der Visualisierung noch Berechnungen für das haptische Rendern durchgeführt werden sollen, darf die Performancelastung nicht zu groß werden. Hierbei muss auf die typische Computerausstattung der Forscher im Bereich der strukturellen Biologie Rücksicht genommen werden. Es sind normale Workstations verbreitet, auf deren Leistungsfähigkeit die hier vorgestellte Lösung abgestimmt sein muss.

Neben der Übersichtlichkeit und der Geschwindigkeit sollte es möglich sein, bestimmte Bestandteile der Moleküle durch eine besondere Darstellungsweise zu betonen, andere Bestandteile wiederum eher unauffällig zu zeichnen. Aus diesen Anforderungen ergibt sich der Wunsch nach großer Flexibilität und Konfigurierbarkeit bei der Darstellung der PDB Dateien, welche im Verlauf der Arbeit noch genauer herausgearbeitet wird.

Neben den Strukturdaten sind ebenfalls die niedrigaufgelösten Elektronenmikroskopie-Dichteverteilungen zu visualisieren. Diese Daten sind im Gegensatz zu den PDB Dateien Volumendaten, für welche die heute verbreitete, auf Oberflächendarstellung optimierte Graphikhardware leider nur unzureichende Visualisierungsmöglichkeiten anbietet. Daher werden im Verlauf der Arbeit verschiedene Darstellungstechniken vorgestellt und verglichen.

Während des Einpassprozesses müssen dreidimensionale Objekte im virtuellen Raum angeordnet werden. Normalerweise steht hierzu aber nur ein zweidimensionaler Computermonitor und ein zweidimensionales Eingabegerät, die Computermaus, zur Verfügung. Diese Geräte stellen keinen praktikablen Zugang zu einer dreidimensionalen virtuellen Welt zur Verfügung. Erst durch den Einsatz von Virtual-Reality-Systemen kann der Umgang mit den virtuellen Objekten intuitiver gestaltet und die Orientierung im dreidimensionalen Raum erleichtert werden. Im Verlauf der Arbeit wird das Forschungsgebiet vorgestellt und typische Bestandteile eines VR-Systems beschrieben.

1.3 SenSitus

Im Rahmen der vorliegenden Arbeit wurde ein Programm mit dem Namen "SenSitus" entwickelt, welches die oben theoretisch beschriebenen Eigenschaften praktisch umsetzt.

Dieses Programm und seine Realisierung werden in der Arbeit ebenfalls beschrieben. Bei der Implementierung wurden verschiedene grundlegende Programmmodule realisiert, die nicht nur in SenSitus eingesetzt werden können. Diese Module sind unter dem Namen “Scientific Visualization Toolkit” (kurz SVT) zusammengefasst worden und werden übersichtsartig beschrieben. Ebenso wurde eine Bibliothek zur Ansteuerung von Virtual-Reality-Eingabegeräten und insbesondere Krafterückkopplungsgeräten entwickelt. Diese Bibliothek wird ebenfalls kurz vorgestellt.

Am Ende der Arbeit werden die Ergebnisse sowie in der Zukunft mögliche Verbesserungen diskutiert.

Kapitel 2

Visualisierung und Virtual Reality

Dieses Kapitel stellt die für diese Arbeit grundlegenden Themengebiete Visualisierung und Virtual Reality vor. Es werden heute per Experiment oder per Simulation große Datenmengen ermittelt. Diese Daten sind zunächst bedeutungslos, erst durch das Aufzeigen der hinter ihnen verborgenen Zusammenhänge werden sie zu einem Schlüsselement der Forschung. Diese Zusammenhänge zu ermitteln ist oft ein langwieriger Prozess, in dessen Verlauf die Visualisierung heute eine wichtige Rolle spielt.

Neben der Visualisierung existiert das verwandte Forschungsgebiet “Virtual Reality” (kurz VR). Es erlaubt einen intuitiveren Umgang mit dem Computer, indem es sich bei der Mensch-Maschine-Interaktion an natürlichen Interaktionsmustern orientiert. Dadurch wird die Analyse komplexer Datensätze ebenfalls wesentlich erleichtert, wodurch VR eine natürliche Ergänzung zur Visualisierung darstellt. Im Verlauf des Kapitels wird in Abschnitt 2.2 der Begriff “Virtual Reality” und das Forschungsgebiet vorgestellt.

2.1 Visualisierung

“Richard Hamming observed many years ago that ‘The purpose of [scientific] computing is insight, not numbers.’ The goal of visualization is to leverage existing scientific methods by providing new scientific insight through visual methods.”

McCormick et.al. [38]

McCormick beschreibt hier allgemein die Rolle der Visualisierung. Nach Richard Hamming sei das Ziel des Scientific Computing Einsicht und nicht die Erzeugung von Zahlen. Das Ziel von Visualisierung sei es, wissenschaftliche Methoden voranzubringen, indem neue Einsichten durch visuelle Techniken ermöglicht werden.

Später beschreibt er das gleiche Problem als “information without interpretation”, also Information ohne Interpretation. Zahlen haben nur dann Bedeutung, wenn es gelingt, sie

zu interpretieren. Ein Beispiel ist die Visualisierung einer Molekülbeschreibung in Abbildung 2.1. Die Interpretation der Beschreibung in Form einer Tabelle von fällt offensichtlich schwerer als die der visuellen Repräsentation der gleichen Daten in der rechten Hälfte der Abbildung.

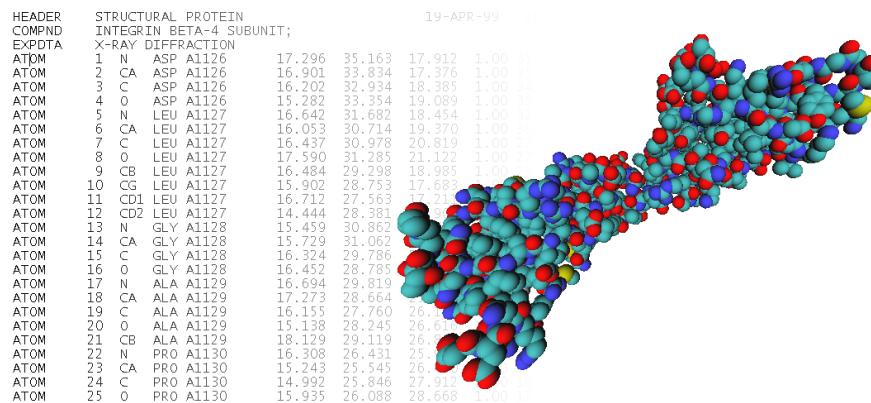


Abbildung 2.1: Visualisierung einer Molekülbeschreibung mit SenSitus

Warum ist dabei die visuelle Darstellung so wichtig? McCormick schreibt hierzu:

“An estimated 50 percent of the brain’s neurons are associated with vision. Visualization in scientific computing aims to put that neurological machinery to work.”

McCormick et.al. [38]

Es wird geschätzt, dass 50 Prozent aller Neuronen im menschlichen Gehirn mit dem Vorgang des Sehens verbunden sind. Diese Schätzung macht deutlich, wie wichtig Sehen für den Menschen und seine Denkprozesse ist.

Visualisierung kann in unterschiedlichen Problemstellungen eingesetzt werden. Steht noch kein Wissen über die hinter den zu visualisierenden Daten stehenden Zusammenhänge zur Verfügung, so ist die Aufgabe der Visualisierung explorativer Natur. Oft existieren Hypothesen, und die Aufgabe der Visualisierung ist, Anhaltspunkte für die Bestätigung oder die Widerlegung solcher Hypothesen zu finden. Eine weitere wichtige Aufgabe für die Visualisierung stellt sich, wenn sich diese Hypothesen bestätigt haben und nun die Ergebnisse der Arbeit Dritten vermittelt werden sollen.

Konkret beschäftigt sich Visualisierung mit der graphischen Darstellung von Datensätzen. Die Darstellungsmöglichkeiten hängen hierbei von der Art der Daten ab. Diese können in verschiedene Kategorien eingeteilt werden, beispielsweise in statische oder zeitvariante Daten, skalare oder vektorielle Daten. Für viele dieser Kategorien haben sich Forschungsbereiche gebildet, in denen spezielle Visualisierungsmethoden entwickelt werden.

2.1.1 Visualisierung von Molekülstrukturdaten

Die Strukturdaten machen einen wesentlichen Teil der in dieser Arbeit zu visualisierenden Daten aus. Im folgenden werden daher einige Möglichkeiten zur Darstellung von Molekülen vorgestellt.

Schon in den 1950er Jahren wurden verschiedene Techniken entwickelt, Moleküle mit Hilfe von realen Drahtmodellen zu visualisieren. Es zeigte sich jedoch bald, dass dieser Ansatz sowohl mechanische Grenzen hatte, als auch, dass er nicht flexibel genug war. Änderungen an diesen komplexen Modellen sind kaum durchführbar, was vor allem im Prozess der Strukturermittlung stört.

Die Entwicklung des Computers hat wesentlich erweiterte Möglichkeiten geschaffen. Schon die ersten grafikfähigen Rechner wurden zur Molekülvisualisierung genutzt, vor allem weil Änderungen der Struktur am Computer leichter zu realisieren sind als an einem echten Modell. Auch konnte mit besseren Graphikcomputern die Darstellung immer weiter verfeinert werden. Heute existiert eine Vielzahl an Molekülvisualisierungsprogrammen, wie zum Beispiel VMD und RasMol, die nachfolgend beschrieben werden.

Moleküle bestehen aus einer Folge von Atomen, die verschiedene Eigenschaften besitzen. Diese Eigenschaften werden für die Visualisierung des Moleküls genutzt. Die Position des Atoms ist hierbei sehr wichtig, zumeist stehen aber noch weitere Informationen zur Verfügung. Das chemische Element, die Ladung, die Zugehörigkeit zu einer Unterstruktur sind weitere Eigenschaften eines Atoms. Aus diesen Informationen können verschiedene Visualisierungen eines Moleküls abgeleitet werden. In der Praxis sind zwei Eigenschaften besonders wichtig, aus denen die meisten vorkommenden Darstellungsarten hergeleitet werden können.

Van der Waals Radius

In vielen Veröffentlichungen werden die Atome eines Moleküls als Kugeln dargestellt. Diese Kugeln werden an den Raumpositionen der Atome gezeichnet und mit Hilfe des sogenannten van der Waals Radius des Atoms skaliert (kurz VDW-Kugeln oder VDW-Darstellung).

Der van der Waals Radius beschreibt eine Eigenschaft von Atomen eines gewissen Typs. Werden zwei Atome aufeinander zu bewegt, so ziehen sich beide an. Die Summe der van der Waals Radien beider Atome gibt den Punkt an, an dem das Potential zwischen den Atomen das Minimum erreicht. Wird die Distanz zwischen beiden Atomen noch weiter verringert, so entsteht eine stark abstossende Kraft.

Der van der Waals Radius wird für ungebundene Atome angegeben. Sind zwei Atome gebunden, so haben ihre Atomkerne eine kleinere Distanz. Jeder Atomtyp besitzt einen Kovalenzradius, der der halben Distanz zwischen zwei gebundenen Atomen des identischen

Typs entspricht. Da zwei gebundene Atome eine kleinere Distanz haben, ist der Kovalenzradius kleiner als der van der Waals Radius.

Werden die Atome eines Moleküls als VDW Kugeln dargestellt, so bietet sich in der Regel eine Skalierung des van der Waals Radius mit einem Faktor kleiner als Eins an. Die gebundenen Atome haben nur eine Distanz von der Summe ihres Kovalenzradius, so dass sich Kugeln mit dem van der Waals Radius überschneiden würden. Wird der Radius mit Eins skaliert, so entsteht eine sogenannte van der Waals Oberfläche.

Atombindungen

Zum Abspeichern der Moleküle hat sich das PDB Dateiformat weit verbreitet. Das PDB Format ist ein zeilenorientiertes, textbasiertes Dateiformat. Jede Zeile besitzt einen Typ, der die in der Zeile gespeicherte Information spezifiziert. Viele PDB Dateien bestehen im wesentlichen nur aus Zeilen des ATOM Typs, die jeweils die Informationen über ein Atom (zum Beispiel Position und chemisches Element) speichern.

Eine wichtige Information neben der Position der Atome stellen die Bindungen unter den Atomen dar. Meist kann die Bindungsinformation nicht aus den PDB Dateien entnommen werden, so dass auf der Grundlage von verschiedenen Kriterien aus den Atompositionen die Bindungen rekonstruiert werden müssen.

Das wichtigste Kriterium stellt hierbei die Entfernung dar. Es wird eine Bindung angenommen, wenn die Distanz zwischen zwei Atomen kleiner als die Summe der van der Waals Radien ist. Ein Problem bei der Rekonstruktion der Bindungen stellt jedoch die oft große Zahl an Atomen in einem Molekül dar. Es müssen die beschriebenen Kriterien zwischen allen Atomen überprüft werden, so dass bei einem einfachen Ansatz hier eine Komplexität von $O(N^2)$ entsteht. Bei großen Molekülen wird diese hohe Komplexität zu einem Problem.

Darstellungsarten

Aus den van der Waals Radien und den Atombindungen können verschiedene Darstellungsarten abgeleitet werden. In der sogenannten CPK-Darstellung (benannt nach Corey, Pauling und Koltun) werden die Bindungen als Zylinder und die Atome als Kugeln gezeichnet (Abbildung 2.2). Analog wird die Licorice-Darstellung erzeugt. Es wird der Durchmesser der Kugeln genauso groß gewählt wie der Durchmesser der Zylinder, die die Bindungen darstellen. So erscheinen die Enden der Zylinder abgerundet.

Ganz ohne Visualisierung der Bindungen kommt die VDW-Darstellung aus. Hier werden die Atome als Kugeln dargestellt, deren Radien den zugehörigen van der Waals Radien entsprechen. Die sogenannten Connolly Dotted Spheres sind ebenfalls Kugeln mit skalierten van der Waals Radien, die aber als Punktwolken gezeichnet werden.

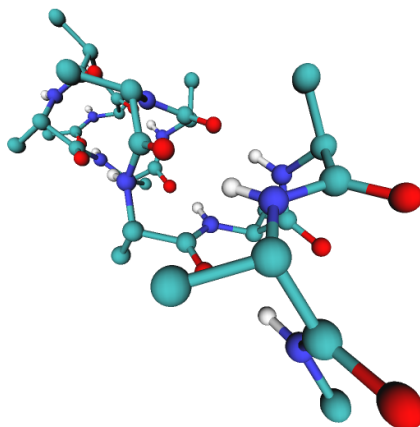


Abbildung 2.2: CPK Darstellung eines Molekül

Darstellungsgeschwindigkeit

Die gerade beschriebenen Visualisierungstechniken setzen Kugeln und Zylinder als Grundkörper für die dreidimensionale Darstellung ein. Diese Grundkörper werden für die Ausgabe durch die Graphikhardware mit Hilfe von Dreiecken angenähert. Um eine realistisch aussehende Kugeldarstellung zu erhalten, werden zirka 256 Dreiecke benötigt. Um ein Molekül mit 50.000 Atomen visualisieren zu können, werden in der VDW-Darstellung $256 * 50.000 = 12.800.000$ Dreiecke benötigt. Eine moderne Graphikkarte kann zirka 200.000 Dreiecke zwanzig mal pro Sekunde zeichnen, so dass das gerade beschriebene Beispiel nicht mehr interaktiv visualisierbar ist.

Wesentlich schneller können weniger aufwendige Grundkörper gezeichnet werden. Werden alle Bindungen als Linien dargestellt und die Atome gar nicht gezeichnet, dann können auch größere Moleküle in Echtzeit visualisiert werden. Diese Darstellung orientiert sich an den Drahtmodellen und ist immer noch ausreichend, um die Struktur des Moleküls zu erfassen.

Vereinfachende Darstellungen

Neben der möglichst kompletten Anzeige aller vorhandenen Informationen ist es auch wichtig, sich bei großen Strukturen nur auf die Darstellung wichtiger Informationen zu beschränken. Große Makromoleküle mit vielen tausend Atomen sind anderenfalls zu unübersichtlich, und Atome im Zentrum können von weiter außen liegenden verdeckt werden.

Meist ist es sinnvoll, sich auf wenige Atomtypen zu beschränken, oder auch nur Teilbereiche der Struktur darzustellen. Neben der Reduktion der Darstellung ist das Mischen von Darstellungsmodi eine sinnvolle Methode zur Lösung des Problems. Wichtige Teilbereiche eines Moleküls können mit aufwendigeren Darstellungsarten hervorgehoben und unwichtigere zum Beispiel mit einer Liniendarstellung weniger auffallend visualisiert werden.

Programme zur Molekülvisualisierung

Es existieren diverse Molekülvisualisierungsprogramme, die sich auf unterschiedliche Anwendungsgebiete spezialisiert haben. RasMol[47] ist eine Entwicklung von Roger Sayle, der als Informatik-Student Anfang der 1990er Jahre eine Molekülvisualisierungssoftware programmierte. Das Programm legt den Schwerpunkt auf eine möglichst schnelle Echtzeitdarstellung der Moleküle. Aus dem später freigegebenen Quellcode wurden andere Programme entwickelt, zum Beispiel Chime[18].

Das Programm VMD [25] — “Visual Molecular Dynamics” — ist eine Entwicklung der Theoretical Biophysics Group [48] und legt den Schwerpunkt auf die Visualisierung von zeitabhängigen Molekulardynamik-Datensätzen.

2.1.2 Visualisierung von Volumendaten

Volumendaten sind eine häufig anzutreffende Datensatzkategorie. Gerade Simulationsprogramme, die auf Hochleistungscomputern ablaufen, erzeugen oft sehr große dreidimensionale Volumendaten. Typisch sind in diesem Bereich aber auch bildgebende Verfahren aus der Medizin oder der Physik, die in Experimenten dreidimensionale Datensätze erzeugen.

Ein dreidimensionaler Volumendatensatz ist ein Skalarfeld $\rho(x, y, z)$. Dieses Feld kann als großer dreidimensionaler Würfel dargestellt werden, in dem jeder Eintrag $\rho(x, y, z)$ als kleiner solider Würfel innerhalb des Volumens gezeichnet wird. Die kleinen Würfel (auch Voxel genannt) werden je nach Wert von $\rho(x, y, z)$ unterschiedlich eingefärbt. Die Einfärbung geschieht mit Hilfe einer sogenannten Transferfunktion, welche die Wertemenge von ρ auf einen Farbraum abbildet. Also zum Beispiel

$$T : \mathbb{R} \rightarrow F \text{ mit } F = \{(r, g, b) | r, g, b \in \{0, \dots, 255\}\}$$

Wird das Skalarfeld auf diese Weise dargestellt, so erkennt der Betrachter aber nur die äußeren Seiten des Würfels, den das Volumen bildet — die außen liegenden Voxel verdecken die weiter innen liegenden Voxel. Dieses Problem kann mit einem erweiterten Farbraum gelöst werden, indem eine zusätzliche “Alpha”-Komponente den Grad der Transparenz angibt:

$$T : \mathbb{R} \rightarrow F \text{ mit } F = \{(r, g, b, a) | r, g, b, a \in \{0, \dots, 255\}\}$$

Auf diese Weise können Voxel transparent gezeichnet werden, so dass auch weiter innen liegende Voxel sichtbar werden. Jedoch hängt dies von der verwendeten Transferfunktion T ab, da sie über den Grad der Transparenz eines Voxels entscheidet. Diese Entscheidung wird hier auf der Grundlage des lokalen Skalarfeldeintrags $\rho(x, y, z)$ getroffen.

Das Aufstellen einer geeigneten Transferfunktion ist eine komplexe Aufgabe, da bei einer ungeeigneten Funktion T Skalarfeldeinträge, die eigentlich für das Verständnis der Daten wichtig wären, transparent und somit nicht sichtbar gezeichnet werden können. Ebenso können bei einer ungeeignet gewählten Funktion nicht interessante Daten wichtige Voxel verdecken. Durch das interaktive Manipulieren der Transferfunktion [32] kann jedoch eine flexible Lösung erreicht werden.

Das Zeichnen der gerade beschriebenen Voxel ist für die heute verbreitete Graphikhardware schwierig. Die Hardware ist meist auf die Darstellung von Oberflächen optimiert, die mit Hilfe von Dreiecken angenähert werden. Eine direkte Darstellung von Voxeln ist nicht vorgesehen.

Neue Graphikhardware implementiert jedoch oft sogenannte dreidimensionale Texturen, die zum Volumen-Rendering [63] genutzt werden können. Es existieren auch Spezialgraphikkarten die für die Darstellung von Volumen mit Hilfe von Voxeln entwickelt worden sind. Ein Beispiel für solche Graphikkarten ist die VolumePro[41] Serie der Firma Terarecon [56]. Hiermit kann das oben beschriebene Volumen-Rendering direkt umgesetzt werden.

Andere Arbeiten versuchen Standardgraphikhardware für die Darstellung von Volumendaten einzusetzen. Lösungen sind hier Oberflächendarstellungen, die für Volumendaten Iso-Oberflächen berechnen [35]. Auch mit Hilfe von zweidimensionalen Texturen, die praktisch jede Graphikhardware hochoptimiert darstellen kann, kann Volumen-Rendering realisiert werden [33].

2.1.3 Visualisierungssoftware

Für die ersten grafikfähigen Computer wurden spezielle Visualisierungsprogramme entwickelt, die nur auf diesen Computern ablauffähig waren. Nachdem immer mehr verschiedene Graphikcomputer verfügbar wurden, stellte sich diese direkte Kopplung als Hindernis heraus. Es etablierten sich verschiedene Softwarebibliotheken, die als Vermittler zwischen der Hardware und dem Anwendungsprogramm fungieren. Beispiele für solche Zwischenbibliotheken sind OpenGL und Direct3D. Sie stellen nur eine abstrakte Graphikschnittstelle zur Verfügung, implementieren aber keine Visualisierungstechniken.

Heute existieren viele Visualisierungsprogramme, die in der Regel nicht mehr an eine spezielle Graphikhardware gebunden sind. Eine allgemeine Softwarebibliothek, mit der Visualisierungsprogramme entwickelt werden können, stellt zum Beispiel VTK dar. VTK startete 1993 als Software zu dem Buch "The Visualization Toolkit: An Object-Oriented Approach

to 3D Graphics". Die Zielsetzung der Autoren Will Schroeder, Ken Martin und Bill Lorensen war, in diesem Buch eine neue objektorientierte Bibliothek vorzustellen, welche eine Visualisierungspipeline realisieren sollte. In diesem System existieren Objekte, welche Daten erzeugen (zum Beispiel indem sie Daten von der Festplatte laden), Objekte, die die Daten verarbeiten (zum Beispiel Filter) und Render-Objekte, welche die Daten anzeigen. Dieses sehr universelle Konzept wird auch in anderen Bibliotheken implementiert, wie zum Beispiel AVS/EXPRESS [4].

Neben diesen allgemeinen Bibliotheken existieren spezielle Visualisierungsprogramme wie die schon vorgestellten Programme VMD und RasMol.

Die vorliegende Arbeit setzt die Bibliothek OpenGL ein, welche daher im folgenden kurz vorgestellt wird.

OpenGL

OpenGL wurde von der Firma SGI entwickelt und ist aus der Bibliothek IrisGL hervorgegangen. War diese Bibliothek noch an die Graphikcomputer der Firma SGI gekoppelt, wurde OpenGL auch auf anderen Systemen verfügbar gemacht. Nachdem sich OpenGL auf anderen Rechner- und Betriebssystemarchitekturen verbreitet hatte, wurde die Weiterentwicklung einer unabhängigen Kommission übertragen, dem "Architecture Review Board", kurz ARB genannt.

OpenGL ist eine klassische, prozedurale Bibliothek, die oft in C programmiert wird. Darüber hinaus existieren für die meisten Programmiersprachen Anbindungen an OpenGL. Die Bibliothek verfügt über eine Zustandsmaschine, welche über eine große Anzahl an Funktionen manipuliert wird. Die Funktionen setzen zum Beispiel Materialeigenschaften wie Farben und Texturen oder Positionen. Die für die Ausgaben zuständigen Funktionen wenden den gerade aktuellen Zustand an. Die Ausgabefunktionen unterstützen nur primitive Grundobjekte, wie Dreiecke oder Polygone. Komplexe Objekte oder Visualisierungsroutinen müssen von dem Anwendungsprogramm selbst aus den Grundobjekten zusammengesetzt werden.

OpenGL konzentriert sich auf das Zeichnen von Graphiken und bietet keine höherwertige Funktionalität an. Daher beinhaltet die Bibliothek keine fertigen Routinen zur Benutzerinteraktion, Kollisionsabfrage oder zur Ereignisbehandlung. Diese Aufgaben werden typischerweise von auf OpenGL aufsetzenden Bibliotheken übernommen.

OpenGL unterstützt nur die Graphikausgabe, nicht aber das Ansprechen von Eingabegeräten oder die Ausgabe von Tönen. Ein Anwendungsprogramm muss für diese Aufgaben weitere Bibliotheken einsetzen. Mit OpenAL [40] existiert für die Tonausgabe eine zu OpenGL passende Bibliothek.

2.2 Virtual Reality

Schon sehr früh hat sich Sutherland 1965 mit der Definition einer virtuellen Realität und später auch mit ihrer konkreten Implementierung beschäftigt.

Den idealen Bildschirm beschreibt er wie folgt:

“The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal.”

Sutherland [52]

Virtual Reality ist eine möglichst realistische Simulation einer Realität mit Hilfe des Computers. Die Simulation soll so realistisch sein, dass sich der Anwender sogar auf einen virtuellen Stuhl setzen können soll, dass eine virtuelle Handschelle ihn wirklich fesselt.

Sutherland gibt aber zu bedenken, dass nicht nur die Nachahmung der Realität Ziel sein sollte, sondern dass in der virtuellen Realität auch ganz andere Regeln gelten könnten:

“There is no reason why the objects displayed by a computer have to follow the ordinary rules of physical reality with which we are familiar”

Sutherland[52]

Als konkretes Beispiel beschreibt er die mögliche Transparenz von Materie in einer virtuellen Realität. Eine transparente Darstellung eines eigentlich nicht durchsichtigen Körpers entspricht zwar nicht der Realität, ermöglicht es aber dem Betrachter, die hinter dem Körper liegenden Objekte wahrzunehmen.

Sutherland ist es wichtig, dass nicht nur die visuelle Darstellung realisiert wird, sondern dass möglichst viele Sinne des Menschen mit virtuellen Eindrücken angesprochen werden:

“.. it [the ultimate display] should serve as many senses as possible”

Sutherland [52]

Neben den Augen und dem Gehör beschreibt er sehr ausführlich die Möglichkeit, für den Tastsinn künstliche Eindrücke zu erzeugen. Das künstliche Hervorrufen eines Tasterindrucks wird “haptisches Rendern” genannt, und wird in Kapitel 3 genauer beschrieben.

Eine exakte, weitgehend akzeptierte Definition des Begriffs “Virtual Reality” (ab hier nur kurz VR) existiert nicht. In der Literatur werden unterschiedliche Definitionen gebraucht, zum Beispiel in [16]:

“A VR-System is one which provides real-time viewer-centered head-tracking perspective with a large angle view, interactive control and binocular display.”

Cruz-Neira [16]

Solche Definitionen legen jedoch den Schwerpunkt eher auf die technische Realisierung als auf allgemeine Grundlagen. Viele Definitionen gehen noch weiter und schreiben ganz konkrete Geräte zur Realisierung einer virtuellen Realität vor. Da zum Beispiel die sogenannten “Head Mounted Devices” (kurz HMD, siehe Abschnitt 2.2.1) schon sehr früh entwickelt worden sind, werden sie von vielen Definitionen als einzige Bildschirm-Technologie vorgesehen. Aus heutiger Sicht erscheint dies nicht sinnvoll.

Es existieren allgemeinere Definitionen, wie zum Beispiel

“Virtual reality is a computer generated simulated environment, which users perceive as real with their natural senses and with which they can interact.”

oder die Definition des Duden:

“Virtuelle Realität: vom Computer simulierte Wirklichkeit, künstliche Welt, in die man sich mit Hilfe der entsprechenden technischen Ausrüstung scheinbar hineinversetzen kann.”

Neben dem Begriff der virtuellen Realität existieren verwandte Begriffe. Der Begriff “Augmented Reality” beschreibt die Möglichkeit, die Realität zu erweitern (augmented = erweitert) und nicht wie bei der Virtual Reality komplett durch eine künstliche Welt zu ersetzen. Dieser Ansatz erlaubt es, die normale Realität um zusätzliche Informationen anzureichern.

Warum VR?

Die Interaktion eines Menschen mit einem Computer unterscheidet sich deutlich von seiner Interaktion mit der Realität. Reale Gegenstände werden direkt mit der Hand berührt und manipuliert. Virtuelle Gegenstände die auf einem Computerbildschirm dargestellt werden, müssen indirekt mit einem Eingabegerät verändert werden. Informationen werden dem Computer per Tastatur übermittelt und nicht wie sonst in der Realität mit Hilfe von gesprochenen Worten.

Je stärker sich die Mensch-Maschine-Interaktion von der Interaktion mit der Realität unterscheidet, desto größer ist der Lernaufwand für einen Computerbenutzer. Setzt ein Mensch zum ersten Mal einen Computer oder ein bestimmtes Programm ein, so benötigt er daher eine lange Einarbeitungszeit. Besonders signifikant weicht die Mensch-Maschine-Interaktion bei dreidimensionalen virtuellen Objekten von den gewohnten Interaktionsmustern ab. Der zweidimensionale Computerbildschirm ist ebenso wie die zweidimensionale Computermouse als Zugang zu einer dreidimensionalen virtuellen Welt nur bedingt geeignet. Gerade bei der Analyse von wissenschaftlichen Datensätzen führt dies dazu, dass der Benutzer die spezifischen Eigenschaften der Daten durch den unintuitiven Umgang nur schwer erfassen kann.

VR versucht mit Hilfe des Computers eine virtuelle Realität zu schaffen, die sich (wenn sich die oben aufgeführten Definitionen später einmal vollkommen realisieren lassen) vom Benutzer wie die Realität manipulieren läßt. VR versucht also, die Mensch-Maschine-Interaktion an die Interaktion des Menschen mit der Realität anzupassen und den Umgang mit dem Computer so intuitiver zu gestalten.

Mit Hilfe eines VR-Systems erhält der Benutzer einen dreidimensionalen visuellen Eindruck von der virtuellen Welt. Zusammen mit wirklich dreidimensionalen Eingabegeräten lassen sich virtuelle Objekte besser untersuchen. Gerade die Analyse komplexer wissenschaftlicher Datensätze profitiert von dem leichteren Zugang zur virtuellen Welt.

2.2.1 VR-Systeme

Ein konkretes VR-System besteht aus verschiedenen Teilkomponenten, welche zum Beispiel von Brooks [11] beschrieben werden. Zum Erzeugen der verschiedenen Sinneseindrücke können jeweils unterschiedliche Komponenten eingesetzt werden.

Zum Erzeugen der visuellen Informationen wird ein Anzeigesystem benötigt. Es besteht in der Regel aus einer Komponente, die das stereoskopische Sehen ermöglicht. Darüber hinaus muss der Computer für die Erzeugung eines realistischen visuellen Eindrucks die aktuelle Betrachterposition ermitteln. Dazu wird ein sogenanntes Headtracking-System eingesetzt. Natürlich wird daneben die eigentliche Anzeigeeinheit benötigt.

Es besteht neben der Anzeige von visuellen Informationen auch die Möglichkeit der Ausgabe von Tonsignalen. Auch hier muss die Betrachterposition bei der Tonausgabe berücksichtigt werden. Da in der vorliegenden Arbeit von dieser Möglichkeit kein Gebrauch gemacht wurde, wird auf eine weitergehende Beschreibung der existierenden Methoden verzichtet.

Mit Hilfe von sogenannten haptischen Geräten ist es darüber hinaus möglich, die Oberfläche von Objekten abzutasten oder einen Krafteindruck (zum Beispiel von dem Gewicht eines Gegenstandes) zu erhalten. Die verschiedenen Geräte und die dort auftretenden Probleme werden in Kapitel 3 beschrieben.

Stereoskopische Darstellung

Ein Betrachter ist bei einem normalen, monoskopischen Monitorbild nicht in der Lage, die korrekte Tiefeninformation zu erfassen. Dies liegt daran, dass beide Augen das gleiche Monitorbild sehen. Betrachtet ein Mensch jedoch eine reale Szene, dann sehen die zwei Augen unterschiedliche Bilder, da sie leicht verschoben im Kopf positioniert sind.

Für die Tiefeninformation ist es wichtig, dass für beide Augen ein jeweils passendes Bild berechnet und dargestellt wird. Ein Computersystem besitzt in der Regel jedoch nur einen Monitor, so dass ohne Zusatzgeräte nur ein monoskopisches Bild dargestellt werden kann.

Wird auf dem Computermonitor abwechselnd das Bild für das rechte und das linke Auge angezeigt, so muss dafür gesorgt werden, dass nur das jeweils angesprochene Auge die Information zu sehen bekommt. Dies erreichen sogenannte “Shutterbrillen”. Diese verdunkeln mit Hilfe von LCD Elementen das gerade nicht angesprochene Auge. Genau in dem Moment, in dem der Monitor auf das Bild für das zweite Auge umschaltet, schaltet auch die Shutterbrille um und verdunkelt das erste Auge. Dieses Umschalten muss sehr schnell erfolgen, soll der Betrachter kein flimmerndes Bild wahrnehmen. Daher wird hier mit Umschaltfrequenzen von mindestens 100 Hertz gearbeitet. Durch das Abdunkeln des gerade nicht angesprochenen Auges nimmt der Betrachter einen leichten Helligkeitsverlust wahr.

Autostereoskopische Monitore versuchen, ohne zusätzliche Brillen auszukommen. Daher werden die Bilder direkt am Monitor durch unterschiedliche Aufsätze so getrennt, dass nur das jeweils angesprochene Auge das Bild auch sehen kann. Durch diese Aufsätze wird meist die Auflösung der Monitore reduziert und die Betrachterposition ist nicht mehr frei wählbar.

Ein weiterer Nachteil aller stereoskopischen Verfahren ist der Performanceverlust. Da die Graphikkarte doppelt so viele Bilder zeichnen muss, halbiert sich die Geschwindigkeit der Graphikausgabe.

Anzeigegeräte

Computermonitore werden als Anzeigegeräte für kleine VR-Systeme eingesetzt. Ihre Größe ist jedoch beschränkt, so dass der Betrachter nur ein kleines “Fenster” zur virtuellen Welt nutzen kann. Werden Shutterbrillen für die stereoskopische Darstellung eingesetzt, so sind in der Regel nur Röhrenmonitore geeignet. Die Anzeigegeräte müssen beim Einsatz von Shutterbrillen mit mindestens 100 Hertz das Bild umschalten können. Flüssigkristall- oder auch Plasmamonitore können diese kurzen Umschaltzeiten beim aktuellen Stand der Technik nicht erreichen.

Projektionssysteme werden eingesetzt, wenn die beschränkte Größe von Monitoren nicht mehr ausreichend ist. Auch hier gilt die Beschränkung beim Einsatz von Shutterbrillen. Die kurzen Umschaltzeiten werden nur von wenigen Projektoren eingehalten (zum Beispiel von klassischen Röhrenprojektoren oder speziellen DLP-Projektoren).

Messen der Betrachterposition

Das Messen der Betrachterposition kann mit verschiedenen Sensortechniken erfolgen. Weit verbreitet sind Systeme, die die Position und Orientierung eines Sensors in einem Magnetfeld messen können. Durch das Magnetfeld sind diese Systeme empfindlich gegenüber metallischen Stoffen, was die Konstruktion des VR-Systems einschränkt.

Optische Systeme nehmen mit Hilfe von verschiedenen Kameras Bilder des Betrachters auf und ermitteln mit Hilfe von Bilderkennungsverfahren seine aktuelle Position. Die Bilderkennungsverfahren müssen hochoptimiert sein, damit die Positionserkennung für ein interaktives Arbeiten schnell genug erfolgt.

Es ist auch möglich, die Betrachterposition mit Hilfe von Ultraschall zu messen, solche Systeme sind aber weniger verbreitet.

HMD

“Head Mounted Devices” (kurz HMD) bestehen aus einem Helm, den der Betrachter aufsetzen muss. Der Helm hat ein spezielles Visier, in welchem zwei kleine Bildschirme untergebracht sind. Jeder Bildschirm ist genau vor einem Auge des Betrachters montiert, so dass er später das für dieses Auge passende Bild anzeigen kann. Durch diese Technik erhält der Betrachter ein stereoskopisches Bild.

Oft werden die HMDs so konstruiert, dass die reale Welt für den Benutzer nicht mehr sichtbar ist. Dadurch erübrigt sich ein absolutes Messen der Kopfposition, nur noch relative Positionsänderungen müssen ermittelt werden. Dies ist wesentlich einfacher, und daher stellen viele HMDs heutzutage eine kostengünstige VR-Komplettlösung dar. Andere HMDs sind so konstruiert, dass der Benutzer die Umwelt noch wahrnehmen kann. Sie bilden meist die Grundlage für Augmented-Reality-Systeme.

Entwickelt wurde das erste HMD von Sutherland schon im Jahre 1968. Der Pionier der VR-Forschung hatte sie für sein erstes VR-System entwickelt, mit der Folge, dass in den darauffolgenden Jahren der Begriff VR und der Begriff HMD oft gleichbedeutend verwendet wurden.

CAVE

In den ersten Jahren der VR-Forschung lag das Hauptproblem auf der Seite der ungenügenden Graphikfähigkeiten der Computer. Anfang der 1990er Jahre zeigte sich jedoch, dass sich die dreidimensionale Computergraphik weiterentwickelt hatte und mittlerweile zu sehr guten Ergebnissen kam. Gleichzeitig wurde offensichtlich, dass Computergraphik alleine noch keine virtuelle Realität ausmacht.

Wie Cruz-Neria et al. in [16] beschreibt, wurde auf einer Panel-Diskussion der Siggraph-Konferenz 1992 die Entwicklung von einem „großen“ VR-System als Alternative zu den HMDs gefordert. Vor allem sollten Forscher aller Disziplinen mit Hilfe des neuen VR-Systems in der Lage sein, ihre Ergebnisse auf Konferenzen zu präsentieren.

Cruz-Neira entwickelte daraufhin mit ihrer Arbeitsgruppe ein VR-System mit mehreren großformatigen Abbildungsflächen. Es entstand ein kleiner Raum, auf dessen Wänden und

Boden dreidimensionale Computergraphik projiziert wurde (siehe Abbildung 2.3). Dies geschah mit Hilfe einer Rückprojektion und großen Röhrenprojektoren. Nur die Boden-graphik wurde mit Hilfe einer Aufprojektion realisiert. Die Decke und die rückwärtige vertikale Wand wurden nicht genutzt.

Bei der Graphikausgabe wurde eine betrachterabhängige Projektion zusammen mit Shutterbrillen für eine stereoskopische Darstellung eingesetzt. Dadurch war der Benutzer in der Lage, virtuelle Gegenstände als Teil des Raumes wahrzunehmen, er konnte um sie herum gehen und sie sich näher anschauen. Da der Betrachter quasi überall das Computerbild wahrnimmt, entsteht ein immersiver Eindruck der virtuellen Welt.

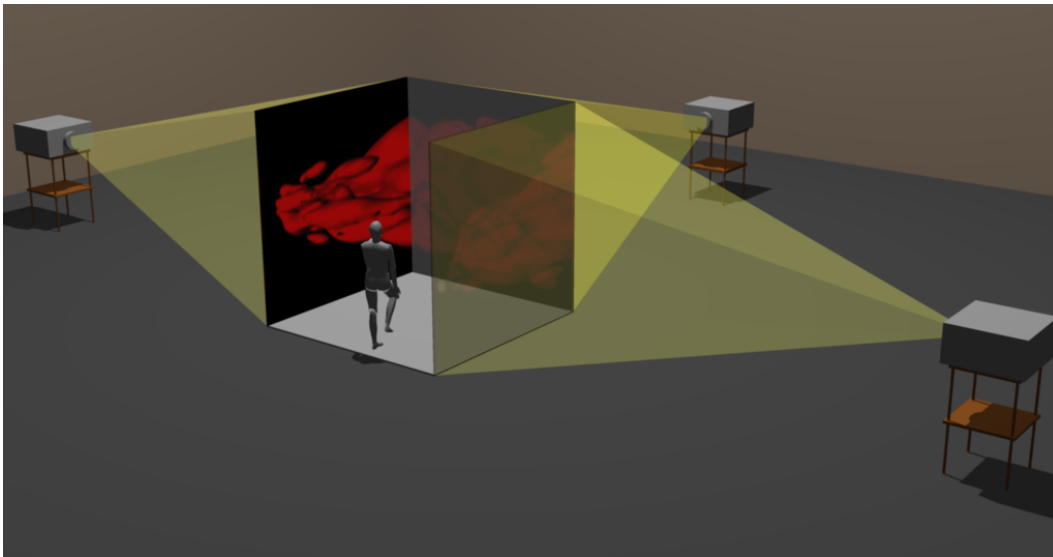


Abbildung 2.3: CAVE

Die Computergraphik wurde mit Hilfe von vier Graphikworkstations realisiert, die miteinander über ein spezielles Hochgeschwindigkeitsnetzwerk gekoppelt waren. Eine fünfte Workstation war für die Abfrage des Headtracking-Systems zuständig.

Die CAVE ist sehr groß und wird entgegen der ursprünglichen Absicht nur selten für die Präsentation auf Konferenzen und Ausstellungen eingesetzt. Die Projektoren und das komplexe Computersystem machen die CAVE zudem zu einer relativ teuren Investition.

VR-Workbench

Neben der CAVE entwickelten sich ähnliche Ansätze, die ebenfalls auf großflächigen Projektionssystemen basieren. Die CAVE erzeugt mit ihren wie ein Raum angeordneten Projektionsleinwänden eine sehr immersive Darstellung der virtuellen Realität. Dies ist für Anwendungen, die nur wenige Datensätze dreidimensional visualisieren und manipulieren wollen, nicht die einzig mögliche Methode.

Ein Beispiel für einen alternativen Ansatz ist die VR-“Werkbank”. Im Gegensatz zur CAVE werden hier nur ein oder zwei Projektionsflächen verwendet, die eine Art Tisch oder Werkbank darstellen. Wird dies mit einer stereoskopischen Darstellung und einer betrachterabhängigen Projektion verknüpft, dann können virtuelle Objekte auf der Werkbank dargestellt und bearbeitet werden.

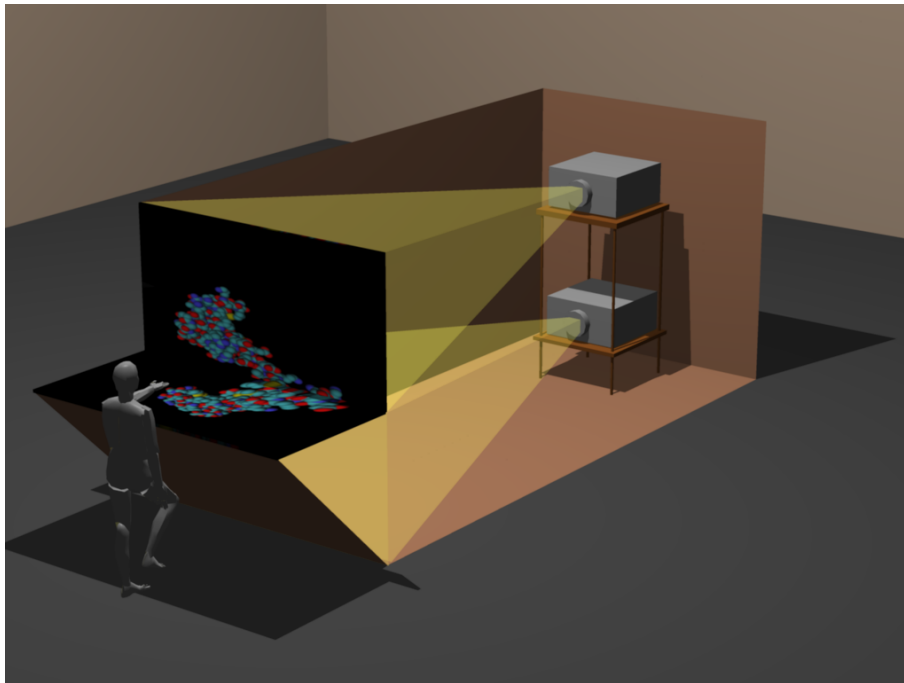


Abbildung 2.4: VR-Workbench oder Holobench

In Abbildung 2.4 ist die für diese Arbeit zur Verfügung stehende, sogenannte “Holobench” des Zentralinstitutes für Angewandte Mathematik des Forschungszentrums Jülich zu sehen. Das System besitzt zwei senkrecht zueinander stehende Projektionsflächen, die durch rückseitig angebrachte Projektoren betrieben werden. Das Bild für die untere Projektionsfläche wird mit Hilfe eines Spiegels passend abgelenkt, so dass beide Projektoren übereinander montiert werden können. Eingesetzt wird ebenfalls ein Headtracking-System sowie Shutterbrillen für die stereoskopische Darstellung.

Responsive Workbench

Sollen mehrere Forscher gleichzeitig mit der CAVE oder der Holobench arbeiten, um zum Beispiel über die visualisierten Datensätze zu diskutieren, so ist dies nur eingeschränkt möglich. Durch die betrachterzentrierte Darstellung erhält nur ein Benutzer ein korrektes Bild, für alle anderen Betrachter erscheint das Bild verzerrt.

Agrawala et.al. [1] schlagen hierzu eine Lösung vor, mit deren Hilfe zwei Personen gleichzeitig an einer “Responsive Workbench” (ein VR-System mit einer Anzeigefläche in einer

Tischplatte) arbeiten können.

Hierzu werden die Kopfpositionen beider Benutzer laufend gemessen. Um für jeden Betrachter und für jedes Auge das richtige Bild zu generieren und zu trennen, werden Shutterbrillen eingesetzt. Zunächst wird das Bild für das linke Auge des ersten Betrachters dargestellt. Das rechte Auge des ersten Betrachters und beide Augen des zweiten Betrachters werden von den Shutterbrillen abgedunkelt. Danach wird das Bild für das rechte Auge des ersten Betrachters dargestellt und dieses Auge von seiner Shutterbrille freigegeben. Danach fährt das System mit den Darstellungen für den zweiten Betrachter fort.

Die Shutterbrillen müssen also um einen neuen Zustand "beide Augen abgedunkelt" erweitert werden. Weiterhin muss die Synchronisation zwischen den Brillen und dem Anzeigesystem so realisiert werden, dass alle vier Zustände (Betrachter A - linkes Auge, A - rechtes Auge, B - linkes Auge, B - rechtes Auge) korrekt dargestellt werden.

Mit Hilfe dieses Systems können die Benutzer über die Eigenschaften der dreidimensionalen Gegenstände diskutieren, da beide die virtuellen Gegenstände an der gleichen Position und mit der gleichen Größe sehen. Die Autoren schlagen jedoch auch vor, dass die virtuelle Szene nicht für beide Betrachter gleich dargestellt wird. Zum Beispiel könnte ein Lehrer Lösungen zu Aufgaben eingeblendet bekommen, der Schüler jedoch nicht.

Durch die vier Phasen und die längeren Zeiten, in denen die Augen der Betrachter abgedunkelt sind, ist der Helligkeitsverlust größer als bei der Nutzung von Shutterbrillen durch eine Person. Ausserdem sinkt die Performance der Darstellung weiter deutlich ab, da nun vier Bilder für einen Bildaufbau gezeichnet werden müssen.

VR-Computersysteme

Bei der Beschreibung der VR-Systeme wurde noch nicht auf die benötigten Computersysteme eingegangen. Da viele VR-Systeme über mehr als eine Anzeigeeinheit verfügen, müssen die Computersysteme mehrere Graphikeinheiten besitzen, um diese anzusteuern. Weiterhin wird für das Darstellen der dreidimensionalen Szenen eine hohe Rechenleistung benötigt, über die in der Regel nur Parallelrechnersysteme verfügen. Es genügen in der Regel jedoch Systeme mit wenigen Prozessoren, Massivparallelrechnersysteme werden nicht benötigt.

Oft werden Shared-Memory-Parallelrechner mit mehreren Graphikeinheiten eingesetzt. Diese Rechner gewährleisten durch die Kopplung der Prozessoren über den Speicher eine sehr schnelle Kommunikation und benötigen keine aufwendige Datensynchronisation, da alle Prozessoren immer auf die aktuellen Werte zugreifen können. Zudem können die Prozessoren die Pipelines der Graphikeinheiten parallel mit Daten versorgen, was gegenüber Einzelprozessorsystemen eine Performancesteigerung bewirkt.

Als preiswerte Alternative werden Cluster-Systeme (zum Beispiel im WireGL/Chromium Projekt [26]) eingesetzt. Die Prozessoren besitzen hier eigenen Speicher und eine eigene

Graphikeinheit, sind untereinander aber nur durch ein relativ langsames Netzwerk verbunden. Gegenüber den Shared-Memory-Parallelrechnern ist dies vor allem durch die anfallenden Synchronisationsprozesse ein Performance-Engpass. Wird die Szene verändert, so müssen die Veränderungen zu allen Prozessoren des Clusters über das Netzwerk übertragen werden.

2.2.2 VR-Software

Durch die hohen Investitionen, die ein VR-System benötigt, haben sich diese Systeme nur relativ wenig verbreitet. Für Softwarehersteller bedeutet dies, zusammen mit der heterogenen Hardwareausstattung, dass nur ein kleiner Markt für ihre Produkte vorhanden ist. Verglichen mit dem Marktsegment der Visualisierungssoftware existiert daher nur wenig kommerzielle Software.

Es existieren kaum Anwendungsprogramme, dafür aber einige Softwarebibliotheken, mit deren Hilfe VR-Programme erstellt werden können. Die CAVELIB [61] des Herstellers VR-CO ist eine solche Basisbibliothek, mit der Anwendungen für VR-Systeme wie die CAVE entwickelt werden können. Ebenso bietet der Hersteller EAI die Bibliothek WorldToolkit (kurz WTK) als Grundlage für VR-Anwendungen an. Die VR-Version AVS/MPE des verbreiteten Visualisierungssystems AVS/EXPRESS kann ebenfalls in der CAVE oder der Holobench eingesetzt werden.

Neben diesen kommerziellen Systemen existiert eine große Anzahl an wissenschaftlichen Softwarebibliotheken und Anwendungsprogrammen für VR-Systeme. Diese werden in der Regel nicht verkauft, sondern oft kostenfrei zur Verfügung gestellt.

VISTA [43] ist eine solche universitäre Entwicklung des Rechenzentrums der RWTH Aachen, die eine Basisbibliothek für VR-Anwendungen darstellt. Die benötigte Funktionalität, sei es das Ansteuern von Eingabegeräten, von Multi-Display-Systemen oder die Bereitstellung eines Szenengraphen, wird von VISTA angeboten. Durch die Wahl von WTK als Implementierungsbasis konnte die Entwicklung von VISTA auf die Kernbereiche, wie zum Beispiel das Ansteuern von VR-Systemen wie einer Holobench oder einer CAVE, konzentriert werden. Dem Anwendungsprogrammierer stehen darüber hinaus Komponenten zur physikalisch-basierten Modellierung oder auch eine komplette dreidimensionale Benutzeroberflächenbibliothek zur Verfügung. Ein anderes System, mit dem ebenfalls VR-Anwendungen erstellt werden können, ist der VR-Juggler [10], entwickelt von der Gruppe um Cruz-Neira.

Die Bibliothek VRPN [53] stellt eine universelle Schnittstelle zum Ansprechen von Eingabegeräten dar. Sie ist im Rahmen des Grope/nanoManipulator Projektes [7, 54] entstanden, welches später vorgestellt wird. VRPN wurde in C++ implementiert und stellt einen durchgehenden Client-Server-Ansatz dar, so dass die Eingabegeräte auch über das Netzwerk angesprochen werden können. Die Bibliothek und der Sourcecode sind kostenlos erhältlich, der Support erfolgt mittlerweile durch eine Firma.

Kapitel 3

Haptisches Rendern

Dieses Kapitel beschreibt verschiedene Aspekte des haptischen Renderns. Zunächst wird die Terminologie genauer untersucht, bevor kurz auf Grundlagen menschlicher Sinneswahrnehmung eingegangen wird. Um haptisches Rendern in der Praxis einsetzen zu können, werden spezielle Geräte benötigt, welche hier ebenfalls übersichtsartig vorgestellt werden.

Danach wird zunächst der Frage nachgegangen, inwieweit das haptische Rendern den Benutzer eines Computersystems unterstützen kann, bevor die Probleme bei der Realisierung einer konkreten Anwendung skizziert werden. Ebenfalls werden Beispielprojekte, welche haptisches Rendern erfolgreich eingesetzt haben, beschrieben.

3.1 Terminologie

Haptisches Rendern beschreibt eine Methode mit der man „künstlich“ einen taktilen Sinnesindruck hervorrufen kann. Der Begriff „Haptik“ wird von dem griechischen Wort „Haptesthai“ abgeleitet, welches übersetzt „Berühren“ bedeutet. Nun könnte davon ausgegangen werden, dass haptisches Rendern das künstliche Erzeugen eines Tasteindrucks meint, welcher beim Berühren einer Oberfläche entsteht. Grundsätzlich wird in der Literatur dieser Begriff jedoch wesentlich weiter gefasst.

Im allgemeinen werden drei verschiedene Wahrnehmungen zusammengefaßt: haptische, kinesthetische und propriozeptive Rückkopplung.

Haptische Rückkopplung meint die Wahrnehmung, die ein Mensch beim Berühren eines Gegenstandes erfährt. Hierbei sind nur die Nervenzellen aktiv, die direkt unter der Haut liegen und vor allem in der Hand mit einer hohen Dichte vorhanden sind. Diese Nervenzellen können sehr schnell auf Reize reagieren. Sie reagieren auf kleinste Verschiebungen in

der Haut, die beim Abtasten einer Oberfläche entstehen. Ausserdem sind sie in der Lage, Temperaturänderungen zu spüren.

Kinesthetische Rückkopplung meint die Wahrnehmung des Zustandes von Muskeln. Übt ein Mensch eine Kraft auf einen Gegenstand aus, so spannt er seine Muskeln an. Diese Veränderung der Muskeln wird an das Gehirn weitergemeldet. Ebenso werden auf den Körper einwirkende Kräfte über die kinesthetische Rückkopplung ans Gehirn weitergeleitet. Oft wird dieser Begriff gleichbedeutend zu propriozeptiver Rückkopplung benutzt.

Propriozeption beschreibt die Wahrnehmung von Position, Bewegung und Kraft. Hiermit ist gemeint, dass der Mensch seine eigene Position (auch ohne visuelle Information) über die propriozeptive Rückkopplung wahrnimmt. Genauso wird auch die Bewegung und ein Krafteindruck wahrgenommen. Wenn zum Beispiel ein schwerer Gegenstand angehoben wird, so erfährt die Person die Schwerkraft über die propriozeptive Rückkopplung.

Auch die mit dem haptischen Rendern in Zusammenhang stehenden Begriffe werden so weit gefasst. Ein haptisches Gerät ist zum Beispiel nicht unbedingt ein Gerät, welches nur eine haptische Rückkopplung auslösen kann. Auch Kraftrückkopplungsgeräte, welche eine Kraft ausüben und somit eher kinesthetische Geräte sind, werden in der Literatur als haptische Geräte bezeichnet.

3.2 Wahrnehmung

Nachdem der Begriff des haptischen Renderns beschrieben worden ist, stellt sich nun die Frage, wie der menschliche Körper diese haptische, kinesthetische und propriozeptive Rückkopplung erreicht, und welche Eigenschaften diese Abläufe besitzen.

Grundsätzlich wird ein Stimulus von einem Rezeptor wahrgenommen. Wenn der Stimulus eine gewisse Stärke überschritten hat, dann gibt der Rezeptor einen Impuls ins menschliche Netzwerk der Nervenzellen ab. Dort wird dieser Impuls über verschiedene Nervenzellen bis zum Gehirn transportiert, wo der Sinnesreiz registriert wird.

3.2.1 Tastsinn

Der Tastsinn arbeitet mit verschiedenen Rezeptoren, welche innerhalb der Haut liegen. Sie teilen sich in verschiedene Rezeptorarten auf, die unterschiedliche Aufgaben wahrnehmen. Die Meissner-Tastkörperchen sitzen direkt unter der Epidermis und nehmen Druck wahr. Die Ruffini-Körperchen sind im Gegensatz dazu auch für die Temperaturwahrnehmung zuständig und sind in einer etwas tieferen Hautschicht angesiedelt.

Die Rezeptoren sind unterschiedlich schnell adaptierend. Regt ein konstanter Stimulus einen Rezeptor an, so wird er mit einer Folge von Impulsen antworten. Die Rate dieser

Impulse wird mit der Zeit immer kleiner, der Rezeptor “gewöhnt” sich an den Stimulus. Sehr langsam adaptierende Rezeptoren, wie zum Beispiel die Ruffini-Körperchen, antworten auf einen konstanten Stimulus mit einer fast konstanten Impulsrate. Schnell adaptierende Rezeptoren wie die Meissner-Körperchen nehmen den Stimulus schon nach einer kurzen Zeit nicht mehr wahr. Dies ist in vielen Fällen sinnvoll, sonst würde zum Beispiel das Tragen von Kleidung permanent Sinneswahrnehmungen auslösen.

Eine andere Eigenschaft ist die räumliche Auflösung der Rezeptoren. Einige Rezeptoren besitzen ein sehr kleines rezeptives Feld, also eine kleine Fläche, auf der sie einen Stimulus wahrnehmen. Die Meissner-Körperchen gehören in diese Kategorie, weshalb sie in einer sehr hohen Anzahl in der Haut unserer Finger existieren. Dadurch besitzt unsere Druckwahrnehmung dort eine besonders hohe räumliche Auflösung. Im Gegensatz dazu besitzen die Ruffini-Körperchen ein großes rezeptives Feld, und unsere Temperaturwahrnehmung besitzt demzufolge eine geringere räumliche Auflösung.

3.2.2 Propriozeption

Über die propriozeptive Rückkopplung nimmt der menschliche Körper seine eigene Position und Lage sowie die seiner Extremitäten wahr. Ebenfalls wird die Veränderung von Position und Lage gemessen und so ein Gefühl für Geschwindigkeit erzeugt. Die Position wird über Rezeptoren gemessen, welche sich in den Gelenken befinden. Über diese Rezeptoren kann der Körper den Winkel von Extremitäten zueinander und somit die Position wahrnehmen.

Damit der Mensch seine Hand kontrolliert einsetzen kann, muss die Lage derselben genau gemessen werden. Dazu müssen die Rezeptoren die Winkel der für die Positionierung der Hand entscheidenden Gelenke mit einer ausreichenden Genauigkeit erfassen. Eine Ungenauigkeit bei der Winkelmessung im Schultergelenk wirkt sich auf den Fehler in der Positionsbestimmung der Hand viel stärker aus als zum Beispiel ein Meßfehler im Handgelenk. Untersuchungen [12] haben dies auch praktisch belegt (durchschnittlicher Winkelfehler im Schultergelenk von 0.8° , verglichen mit 2.0° im Handgelenk).

Mit Hilfe der kinesthetischen Rückkopplung kann der Mensch auf ihn einwirkende sowie von ihm erzeugte Kräfte messen. Dies geschieht mit Hilfe von Golgi-Sehnenorganen sowie sogenannten Muskelspindeln.

Die Golgi-Organen befinden sich zwischen den Muskelenden und den Sehnen. Hier können sie die Spannung, die der Muskel erzeugt, beziehungsweise die auf ihn einwirkt, messen. Den Zustand (gedehnt oder kontrahiert) können sie jedoch nicht wahrnehmen, dazu dienen die Muskelspindeln. Diese winden sich um einzelne Muskelfasern und können so den Zustand und die Zustandsänderungen über die Zeit messen.

Obwohl die Funktionen der einzelnen Rezeptoren unterschieden werden können, ist für die konkrete Wahrnehmung im Gehirn die Kombination von allen Rezeptoren ausschlag-

gebend. Wirkt zum Beispiel eine externe Kraft auf den menschlichen Körper, so wird nicht nur über die Muskelspindeln und Golgi-Organen ein Impuls ans Gehirn weitergeleitet, sondern in der Regel auch ein Druck auf die Haut über die Meissner-Tastkörperchen erkannt. Ist der Druck stark, so werden die Rezeptoren in den Gelenken auch noch eine Winkelveränderung feststellen. Erst aus allen Signalen zusammen erzeugt unser Gehirn die Wahrnehmung der Kraft.

3.2.3 Regelung der Muskelaktivität

Die gerade beschriebenen sensorischen Systeme sind ebenfalls für die Regelung der Muskelaktivität von großer Bedeutung. Nur wenn der Körper die Aktivität der Muskeln laufend kontrolliert und immer wieder anpasst, kann der Mensch auch komplexe Aufgaben vollführen.

Wird zum Beispiel ein fragiler Gegenstand aufgehoben, so darf nur so viel Kraft auf ihn ausgeübt werden, dass er nicht zerdrückt wird. Gleichzeitig soll er nicht zu Boden fallen, daher muss eine gewisse Mindestkraft aufgebracht werden. Genau zwischen diesen Grenzen muss sich die tatsächlich erzeugte Muskelkraft bewegen, soll dieser Gegenstand nicht zerstört werden.

Srinivasan [50] hat die Leistung dieser Regelung untersucht. Es wurde unter anderem die Fragestellung bearbeitet, wie präzise ein Mensch auf eine Kraft reagieren kann. Dazu sollten Probanden mit dem Finger eine möglichst exakt gleichgroße Gegenkraft zu einer externen Kraft aufbauen. Diese externe Kraft war sowohl von konstanter als auch veränderlicher Größe. Es zeigte sich, dass sich der Fehler mit ansteigender Kraftgröße linear erhöhte.

Eine andere für das haptische Rendern besonders wichtige Fragestellung ist die Geschwindigkeit, in der Sensoren Stimuli erkennen, und die Geschwindigkeit, mit der die Regelung dann reagieren kann. Generell kann hier festgestellt werden, dass der Mensch Stimuli viel schneller erkennen (20 bis 10000 Hz) als darauf reagieren (5 bis 10 Hz) kann. Die Geschwindigkeit der Erkennung von Stimuli ist jedoch je nach Situation höchst unterschiedlich. Die Finger erkennen beim Abtasten von Oberflächen Stimuli mit sehr hoher Geschwindigkeit, wohingegen die Erkennungsgeschwindigkeit bei großen Bewegungen des gesamten Körpers wesentlich geringer wird.

3.3 Kraftrückkopplungsgeräte

Für das künstliche Erzeugen eines Stimulus werden spezielle Geräte benötigt. Für die Realisierung eines solchen Gerätes existiert kein einheitliches Konzept, es ist vielmehr in den letzten Jahrzehnten eine große Zahl sehr unterschiedlicher Modelle entworfen worden.

Im folgenden wird ein kurzer Überblick über die historische Entwicklung von Krafrückkopplungsgeräten vorgestellt. Haptische Geräte (also Geräte, welche ausschliesslich den Tastsinn ansprechen) sind wesentlich seltener entwickelt worden und werden in der vorliegenden Arbeit auch nicht eingesetzt. Daher werden sie im folgenden nicht weiter besprochen.

In den 1950er Jahren wurde der “Argonne Remote Manipulator”, kurz ARM, entwickelt, welcher aber nicht für Virtual Reality sondern für Teleoperator-Anwendungen genutzt wurde. Die Idee war hier, dass eine entfernte, nicht unmittelbar erreichbare Umgebung direkt manipuliert werden sollte. Anwendungen sind zum Beispiel der Umgang mit gefährlichen Stoffen, oder Umgebungen wie Tiefsee oder Weltraum, in denen ein Mensch Objekte nicht direkt bearbeiten kann. Der ARM besteht aus einem Handteil, das einem Roboterarm ähnelt und welches der Benutzer bedient, um damit einen Stellvertreter in der entfernten Umgebung zu steuern. Interessanterweise besitzt der ARM schon die Möglichkeit, Krafteindrücke, die der Stellvertreter wahrnimmt, an das Handteil zu melden, welches dann diese Eindrücke an den Benutzer ausgibt. Der ARM ist also schon ein echtes Kraftrückkopplungsgerät.

Später entwickelte sich die Computergraphik und Virtual-Reality-Forschung. Man entdeckte auch hier die Notwendigkeit zur Krafrückkopplung (Sutherland 1965 in [52]). Ende der 1960er Jahre startete Brooks das Projekt GROPE [7] und entwickelte hier eigene Krafrückkopplungsgeräte, wie GROPE-I. Dies war ein „zweidimensionales“ Gerät, das der Benutzer mit einem Finger in einer Ebene steuerte. Durch zwei Motoren konnte das Gerät Kräfte auf den Finger des Benutzers ausgeben. Später übernahm Brooks den ARM des Argonne National Lab (die spätere Version ARM III).

Eine andere Entwicklungsidee verfolgten verschiedene Forschergruppen mit der Entwicklung von Exoskelettgeräten. Diese werden wie ein zweites Skelett über dem Skelett des Benutzers getragen und können mit Hilfe von Motoren Druck auf Gelenkachsen wie zum Beispiel Finger ausüben. Mitte der 1960er Jahre experimentierte eine Forschergruppe des Cornell Aeronautical Laboratory mit dem “Hardiman”-Prototypen. Dieses Gerät stellte ein Exoskelett für praktisch den gesamten menschlichen Körper dar. Die Idee war hier, dass der Mensch durch das Gerät in seinen Bewegungen unterstützt werden und so in der Lage sein sollte, schwerste Objekte zu manipulieren.

Einen kompakteren Ansatz entwickelten Jones und Thousand mit dem “Dextrous Master Manipulator”. Dieser stellt ein Exoskelett nur für die Hand dar, das mit Hilfe von kleinen Zylindern arbeitet. Die Kontraktion der Zylinder kann sowohl gemessen werden (um die Bewegungen der Finger zu ermitteln), als auch ein Luftdruck aufgebaut und so eine Kraft ausgegeben werden.

Später wurden dann „zweidimensionale“ Steuerknüppel entwickelt, die ebenfalls Kräfte ausgeben konnten. Ein frühes System war das Sandpaper-System [39] von Minsky et al. Mit Hilfe von eingebauten Motoren konnten Krafteindrücke modelliert und so Oberflächen von simuliertem Sandpapier erfahren werden.

Mitte der 1990er Jahre wurden dann Seriengeräte speziell für Virtual-Reality-Anwendungen vorgestellt. Vor allem die Phantom-Geräte von Massie/SensAble [37] wurden für wissenschaftliche Anwendungen eingesetzt. Sie werden in den folgenden Abschnitten noch genauer vorgestellt.

3.3.1 Eigenschaften von Krafrückkopplungsgeräten

Damit für den Benutzer ein realistischer Krafteindruck entsteht, muss ein Krafrückkopplungsgerät einige allgemeine Eigenschaften erfüllen:

1. “Freier” virtueller Raum muss sich auch “frei” anfühlen. Dies bedeutet, dass ein Krafrückkopplungsgerät den Benutzer keine Kraft spüren lassen darf, wenn es keine Kraft ausgeben soll. Problematisch ist in diesem Zusammenhang die Masse des Krafrückkopplungsgerätes. Die Trägheit der Masse wird von dem Benutzer wie eine Kraft wahrgenommen, muss er das Gerät oder Teile davon mit seiner normalen Bewegung mittransportieren.
2. Feste Gegenstände müssen sich auch solide anfühlen. Dieses Kriterium ist leicht einzusehen: Feste Oberflächen dürfen von dem Benutzer nicht durchdrungen werden können, ansonsten entsteht der Eindruck einer weichen Oberfläche. Auch dieses Kriterium ist nicht trivial. Der Mensch kann relativ hohe Kräfte auf einen Punkt ausüben, so dass ein Krafrückkopplungsgerät diese Kräfte komplett kompensieren muss. Der Mensch kann mit seinen Fingern bis zu 50 Newton Druck kurzzeitig ausüben. Greift der Mensch mit seiner ganzen Hand einen Gegenstand und drückt ihn zusammen, so kann er sogar bis zu 400 Newton kurzzeitig erzeugen.
3. Geschwindigkeit der Kraftänderungen. Streicht der Benutzer über eine virtuelle Oberfläche, um ihre Struktur wahrzunehmen, so muss die Kraft sehr schnell angepasst werden. Wenn die Kraftinformationen nur langsam verändert werden, dann widerspricht dies sehr schnell den vom Benutzer in der Realität gemachten Erfahrungen und wird daher als unrealistisch empfunden und abgelehnt.

Diese allgemeinen Eigenschaften schliessen sich teilweise gegenseitig aus. Beispielsweise muss ein Gerät den hohen Kräften standhalten, die der Benutzer kurzzeitig erzeugen kann. Dies bedeutet in der Praxis meist eine aufwendige Konstruktion aus Metall. Diese Konstruktion hat ein hohes Eigengewicht, dessen Trägheit den Benutzer behindert. Dies bedeutet, dass keine universellen Geräte existieren, die sich für alle Anwendungen einsetzen lassen.

3.3.2 Auswahl eines Gerätes

Die verschiedenen Geräte besitzen unterschiedlich ausgeprägte Merkmale, die für die Auswahl eines konkreten Gerätes im Hinblick auf eine spezielle Anwendung wichtig sind:

1. Schnittstelle zwischen Benutzer und Gerät (zum Beispiel Exoskelett, Steuerknüppel etc.)
2. Größe des Bewegungsraumes
3. Anzahl der Freiheitsgrade (zum Beispiel nur zwei beim Steuerknüppel, aber sehr viele bei Exoskelettgeräten)
4. Maximale Kraftausgabe

Eine detailliertere Diskussion über die grundlegenden Eigenschaften und Selektionskriterien findet man bei Hayward et al. in [22].

In der vorliegenden Arbeit wurde das Phantom der Firma SensAble als Kraftausgabegerät ausgewählt. Es ist eines der wenigen erhältlichen Geräte, die sich für wissenschaftliche Anwendungen eignen. Zudem weist es eine universelle Mensch-Gerät-Schnittstelle auf, so dass man es gut für selbstentwickelte Anwendungen einsetzen kann. Es ist sowohl mit drei als auch sechs Freiheitsgraden der Kraftausgabe erhältlich und kann relativ hohe Kräfte ausgeben. Dieses Gerät soll im folgenden genauer vorgestellt werden.

3.3.3 SensAble Phantom

Das Phantom wurde 1993 von Thomas Massie am Artificial Intelligence Laboratory des Massachusetts Institute of Technology, im Rahmen seiner Bachelor-Abschlussarbeit entwickelt. Massie und Salisbury beschreiben in [37] die Entwicklung des Gerätes. Mittlerweile hat Massie die Firma SensAble Technologies [49] gegründet und verkauft weiterentwickelte Versionen des Phantom.

Das Design des Phantom ist relativ einfach gehalten. Im Prinzip entspricht das Gerät einem Roboterarm, dessen "Hand" sich jedoch nicht selbstständig bewegt, sondern vom Benutzer geführt wird. Anstelle der beim Roboter angebrachten Greifer besitzt das Phantom einen Stift, der vom Benutzer in der Hand gehalten wird.

Der Benutzer kann diesen Stift dreidimensional durch den Raum bewegen, wobei das Phantom gleichzeitig über die verschiedenen Gelenke die aktuelle Position der Hand exakt bestimmt. In einem Computerprogramm wird in der Regel graphisch ein „Stellvertreterobjekt“ dargestellt, das sich simultan zu den Bewegungen der Hand im virtuellen Raum bewegt. Stößt das Stellvertreterobjekt im virtuellen computersimulierten Raum an ein anderes Objekt, so kann die Software die auftretenden Kräfte berechnen und an die Hand des Benutzers ausgeben. Dazu erzeugt der „Roboterarm“ des Phantom die Kräfte per eingebauter Motoren, die über Seilzüge auf die Gelenke des Arms wirken.

Ein Vorteil dieser Konstruktion ist die exakte Messung der aktuellen Handposition und die schnelle und einfache Erzeugung der Kräfte. Das Phantom ist unabhängig von der benutzenden Person und kann im Gegensatz zu Exoskeleton-Designs sofort benutzt werden.

Ein Nachteil der Phantom-Konstruktion ist jedoch der indirekte Interaktionsmechanismus, welcher praktisch einer dreidimensionalen Maus entspricht. Diese indirekte Mensch-Maschine-Interaktion widerspricht dem ansonsten geforderten intuitiven und direkten Umgang mit virtuellen Objekten (siehe Kapitel 2). Ein weiteres Problem in diesem Zusammenhang stellt der Aktionsradius des Gerätes dar. Es ist nur möglich, ein dreidimensionales Volumen abzutasten, abhängig von der Konstruktion des Roboterarms. Die Firma SensAble bietet aus diesem Grunde verschieden große Modelle an:

Modell	Arbeitsvolumen
Phantom Desktop	16cm x 13cm x 13cm
Phantom Premium 6DOF	19,5cm x 27cm x 37,5cm
Phantom Premium 1.0	13cm x 18cm x 25cm
Phantom Premium 1.5	19,5cm x 27cm x 37,5cm
Phantom Premium 3.0	41cm x 59cm x 84cm

Phantom Desktop und Premium

Nachdem die ersten Prototypen von Massie entworfen waren, entstand schnell der Wunsch nach einem kostengünstigen Gerät, welches andere Forschergruppen kaufen könnten. Aus diesem Grundgedanken heraus entstand das kleinste Phantom-Modell, die Desktop-Variante (Abbildung 3.1). Die Desktop-Variante kann die Bewegung der Hand in allen sechs Freiheitsgraden messen und Kräfte in den drei Translationsrichtungen ausgeben.

Die Premium-Variante ist analog zu der Desktop-Variante aufgebaut, ist jedoch wesentlich größer. Sie ist in verschiedenen Abmessungen erhältlich, deren größte Version den Bewegungsradius eines kompletten Schreibtisches bietet.

Phantom 6DOF

Im Gegensatz zu den vorher genannten Geräten kann das Phantom 6DOF (Abbildung 3.2) auch Drehmomente zusätzlich zu den Translationskräften ausgeben. Die dafür notwendigen Motoren sind jedoch kleiner als die Motoren für die Translationskräfte. Sie sind alle an dem Stift bzw. an dem Gelenk des Stiftes angebracht und dürfen daher nicht zu groß und schwer sein. Daher werden kleinere Motoren verwendet, die nicht genauso hohe Kräfte wie die großen Motoren aufbringen können, so dass die erzeugbaren Drehmomente etwas schwächer ausfallen. SensAble gibt an, dass die dauerhaft erzeugbaren maximalen Translationskräfte 1,8 Newton sind, die Drehkräfte hingegen maximal zwischen 0,048 und 0,188 Newtonmeter betragen. Das Drehmoment wird mit Hilfe eines Pivotpunktes \vec{O} und einer Kraft \vec{F} berechnet, die an einem (relativ zu \vec{O} definierten) Punkt \vec{d} angreift. Dann gilt für das resultierende Drehmoment $\vec{N} = \vec{F} \times \vec{d}$ [Nm]. Das maximale Drehmoment von 0,048 Newtonmetern wird also erreicht, wenn eine Kraft von 0,48 Newton in 10 cm Entfernung



Abbildung 3.1: Phantom Desktop

vom Pivotpunkt anliegt. Die erzeugbaren Drehmomente sind in der Praxis aber trotzdem groß genug, um das realistische Empfinden eines Sinneseindrucks zu gewährleisten.

Weiterhin besteht bei den kleinen Motoren die Gefahr der Zerstörung durch Überlastung, wenn der Benutzer zum Beispiel gegen die ausgegebenen Kräfte drückt. Dieses Problem haben die Entwickler bei SensAble durch zusätzliche Temperatursensoren gelöst, deren Zustand per Software abfragbar ist. Arbeitet der Benutzer gegen die ausgegebenen Kräfte an, so kann der Motor die ihm zur Verfügung stehende Energie nicht in eine Drehbewegung umsetzen. Stattdessen wird die Energie in Wärme umgesetzt, so dass der Motor eventuell durch Überhitzung Schaden nimmt. In der Praxis hat sich gezeigt, dass die Temperatur bei einer normalen Benutzung nicht wesentlich ansteigt.

3.4 Qualität des Haptischen Renderns

Bevor nun auf konkrete Implementierungsstrategien und realisierte Projekte eingegangen wird, stellt sich die Frage nach der Qualität von haptischem Rendern. Kann mit Hilfe des haptischen Renders ein Benutzer eines interaktiven Programms so unterstützt werden, dass er bessere Resultate oder gleich gute Resultate in kürzerer Zeit erzielt?

Da das Forschungsgebiet erst seit relativ kurzer Zeit existiert, wurden bisher nur wenige Untersuchungen durchgeführt, die diese Frage bearbeitet haben. Das verwandte Fachgebiet der Telerobotik hat sich jedoch schon länger mit analogen Fragestellungen beschäftigt.



Abbildung 3.2: Phantom 6DOF

Burdea beschreibt dazu in [12] eine Untersuchung von Hannaford und Wood. Hier wurde Probanden folgende Aufgabe gestellt. Sie sollten per Teleoperation einen Zapfen in eine zugehörige Aufnahme einpassen. Hierbei wurde die Zeit gemessen, die die Probanden mit visueller Information und Krafterückkopplung, ohne Krafterückkopplung und ganz manuell benötigten.

Die Probanden konnten in dieser Untersuchung die Aufgabe manuell, also ganz ohne Einsatz der Telerobotik, in der kürzesten Zeit bewältigen. Dies war auch zu vermuten, da in diesem Fall alle haptischen und visuellen Informationen auf natürliche Weise zur Verfügung standen. Sollte jedoch die Aufgabe mit Hilfe der Telerobotik, mit der visuellen Information aber ohne Krafterückkopplung, durchgeführt werden, so dauerte die Bewältigung der Aufgabe im Durchschnitt viel länger. Wurde nun die Krafterückkopplung aktiviert, so ging die Zeit um 30% zurück.

Ein weiteres Ergebnis der Untersuchung war die Reduzierung der Fehler und der aufgewendeten Kraft. War keine Krafterückkopplung aktiviert, so haben die Probanden wesentlich kraftvoller agiert und haben viel mehr Kollisionen mit dem Zapfen beim Einpassen verursacht. Bei einem empfindlichen Gegenstand hätte dies zur Beschädigung oder Zerstörung geführt.

Burdea beschreibt ebenfalls eine andere Untersuchung von Howe und Kontarinis mit einem analogen Aufbau. Hier wurde jedoch der Einfluss der Aktualisierung der Kraftausgabe gemessen. Es zeigte sich, dass eine Erhöhung der Aktualisierungsrate eine Verringerung der Fehler bewirkte. Bei niedrigen Raten stiegen die Fehler dagegen an. Interessanterweise zeigte die Untersuchung jedoch auch, dass schon eine Rate von nur 2Hz die Fehler von 63% auf 14% absenkte, verglichen mit der Situation ganz ohne Krafterückkopplung.

3.5 Kraftberechnung

Sollen mit Hilfe von Krafrückkopplungsgeräten Kräfte ausgegeben werden, so müssen diese zunächst berechnet werden. In diesem Abschnitt wird beschrieben, welche Kriterien eine Kraftberechnung erfüllen muss, damit der Benutzer einen realistischen Krafteindruck erfährt. Danach werden verschiedene Implementierungsstrategien vorgestellt.

Kraft-Update

Da die Berechnung einer Kraft in der Regel nicht trivial ist, stellt sich schnell die Frage, welche Update-Raten erreicht werden müssen, um noch einen realistischen taktilen Eindruck erzeugen zu können. Beim Neuzeichnen des Bildschirminhalts muss mindestens ein Wert zwischen 15 und 25 Updates pro Sekunde erreicht werden. Oberhalb dieser Schwelle kann das Gehirn die Bilder nicht mehr einzeln wahrnehmen, es verschmelzen alle Bilder zu einer kontinuierlichen Bewegung.

Wie schon in Abschnitt 3.2 beschrieben, erkennen die Rezeptoren Stimuli sehr schnell, so dass haptisches Rendern wesentlich höhere Kraft-Update-Raten erreichen muss. Bei einem Tastvorgang mit der Hand können bis zu 10000 Stimuli pro Sekunde erkannt werden.

Zumeist wird davon ausgegangen, dass mindestens 500-1000 Updates pro Sekunde erreicht werden müssen [14, 39]. Bleibt eine Anwendung deutlich unter dieser Schwelle, so treten verschiedene Irritationen beim Benutzer auf. Es ändern sich die Kräfte nicht schnell genug, was bei Objekten wie simulierten Federn oder Flüssigkeiten dazu führt, dass die künstlich erzeugten Sinneswahrnehmungen den in der Realität gemachten Wahrnehmungen widersprechen.

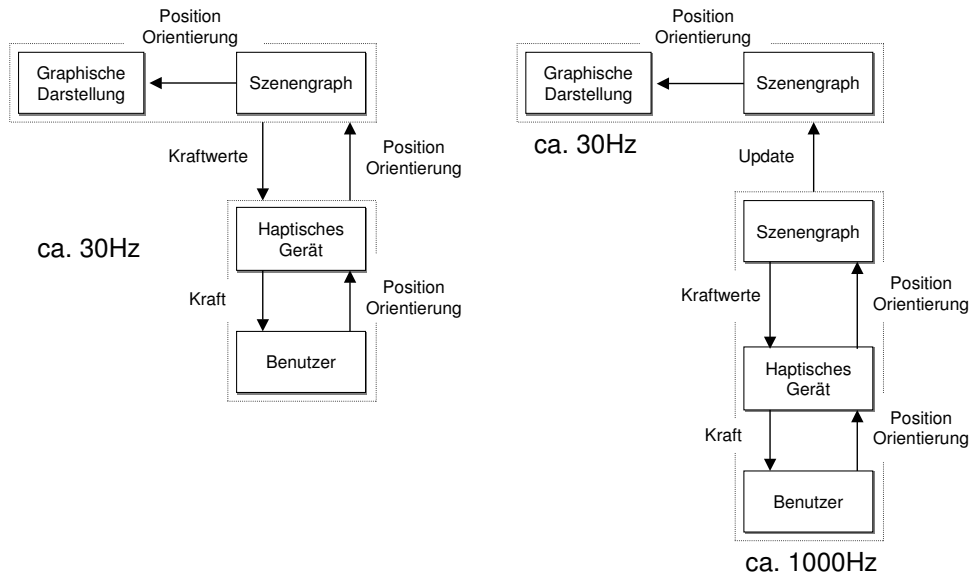
Ein anderes Problem ist, dass Oberflächen bei langsamen Kraft-Update-Raten nicht mehr präzise abgetastet werden können. Der Benutzer kann dann in eigentlich feste Oberflächen eindringen, weil das Programm dies nicht schnell genug bemerkt und erst zu spät eine Gegenkraft aufbaut. Ein realistischer Krafteindruck kann so nicht gewährleistet werden.

Eine mögliche Lösung für dieses Problem stellt die Berechnung von Zwischenwerten dar. Statt einer werden nun zwei verschiedene Kraftberechnungsroutinen eingesetzt, die unterschiedlich schnell arbeiten. Eine präzise Kraftberechnungsroutine erzeugt exakte Werte, aber mit einer zu geringen Frequenz. Eine zweite Routine berechnet aus dem letzten exakten Ergebnis schnell unpräzisere neue Werte, dies aber mit einer ausreichenden Frequenz. Weichen die unpräziseren Werte jedoch zu sehr vom richtigen Kraftwert ab, so treten wieder Irritationen beim Benutzer auf.

Szenengraph

Viele Applikationen nutzen einen Szenengraphen, um die Objekte einer dreidimensionalen Szene zu verwalten. Ein naheliegender Ansatz besteht darin, diesen Szenengraph nicht nur

für das Zeichnen sondern auch für die Kraftberechnung zu verwenden. Zum Zeichnen wird der Szenengraph durchlaufen und nun werden im selben Arbeitsschritt auch die Kräfte für das Kraftrückkopplungsgerät berechnet (Abbildung 3.3 a).



a) Gemeinsamer Szenengraph

b) Zwei Szenengraphen

Abbildung 3.3: Szenengraphen und Haptik

Auf diese Weise ist es einfach, eine bestehende Anwendung um haptisches Rendern zu erweitern, eine größere Veränderung an der Struktur des Programms ist nicht notwendig. Die Kraftberechnung ist nun jedoch an das Zeichnen gekoppelt. So werden nur Kraft-Update-Raten gleich den Bildschirm-Update-Raten erreicht. Diese erreichen normalerweise zwischen 10 und 100 Hz und sind somit deutlich niedriger als die für das haptische Rendern zuvor geforderten 500 Hz - 1 kHz. Dies ist meist zu wenig, um ein realistisches taktilen Empfinden beim Benutzer auszulösen.

Ein anderer Ansatz besteht in der Einführung eines separaten Szenengraphen für die Kraftberechnung. Dieser wird unabhängig von dem Zeichenvorgang mit der geforderten hohen Frequenz durchlaufen (Abbildung 3.3 b).

Dieses Vorgehen bringt aber ebenfalls Nachteile mit sich, die sich in Synchronisationsproblemen äussern. Um diese zu vermeiden, müssen beide Szenengraphen ihren Zustand abgleichen, damit die visuellen Eindrücke zu den taktilen Wahrnehmungen passen. Ausserdem ist es nicht mehr trivial, eine bereits bestehende Anwendung um haptisches Rendern zu ergänzen.

GHOST

GHOST ist eine spezielle Bibliothek der Firma SensAble, um ihre Krafrückkopplungsgeräte anzusteuern. GHOST realisiert einen eigenen Szenengraphen für die Kraftberechnung. Es stehen dem Anwendungsprogrammierer verschiedene Grundobjekte zur Verfügung (zum Beispiel Kugel oder Würfel), aus welchen auch komplexere Objekte zusammengesetzt werden können. Einige wenige komplexere Objektklassen wurden realisiert, wobei hier das universelle Dreiecksgitter hervorzuheben ist.

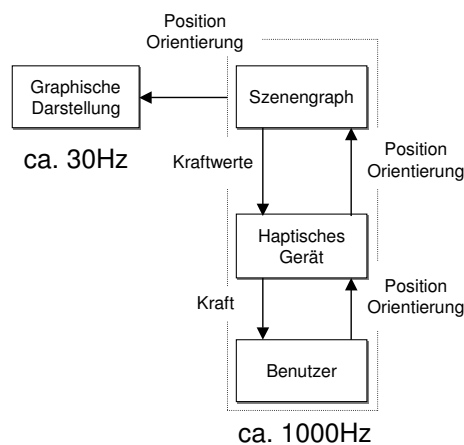


Abbildung 3.4: GHOST

Wird die Hauptschleife dieses Szenengraphen gestartet, so kann der Benutzer die Oberflächen der Objekte abtasten. Gleichzeitig wurde in GHOST auch eine graphische Darstellung der Objekte vorgesehen, so dass hier ein haptischer Szenengraph realisiert wurde, der mit der notwendigen hohen Frequenz aktualisiert wird und der gleichzeitig die visuelle Darstellung zur Verfügung stellt.

Komplexere Anwendungen können bei der in C++ entwickelten Bibliothek eigene Objektklassen implementieren und die Kraftberechnung selbst übernehmen. Auch können Kraftfelder programmiert werden, die gar keine Oberflächen simulieren, sondern Krafteindrücke wie zum Beispiel in Flüssigkeiten.

3.6 Projekte

In diesem Abschnitt werden einige wissenschaftliche Projekte beschrieben, die ebenfalls Krafrückkopplungsgeräte einsetzen bzw. eingesetzt haben.

3.6.1 GROPE

Das GROPE-Projekt [7] der Gruppe um Prof. Brooks von der Universität von Chapel Hill/USA ist eines der ersten Projekte überhaupt, welches Krafrückkopplungsgeräte eingesetzt hat. Das Projekt startete 1967 mit dem Ziel, ein sogenanntes “Haptic Display” für die Darstellung von sechsdimensionalen Kraftfeldern von miteinander interagierenden Molekülen zu entwickeln. Ziel war die Ermittlung von möglichen Positionen für das Docking von Molekülen.

Da 1967 kein Krafrückkopplungsgerät erhältlich war, wurde zunächst ein solches Gerät entwickelt. Mit dessen Hilfe konnten sowohl zweidimensionale Kräfte ausgegeben, als auch Bewegungen der Hand gemessen werden. Das Gerät bestand aus Potentiometern, die die Bewegung eines kleinen Schlittens auf einem flachen Tablett messen konnten. Zusätzlich wurden zwei Motoren angebracht, welche mit Hilfe von kleinen Drahtseilen Kräfte auf den Schlitten ausüben konnten. Der Benutzer konnte mit Hilfe eines Fingers den Schlitten bewegen.

Nach diesem einfachen Design wurde ein wesentlich komplexeres Krafrückkopplungsgerät entwickelt. Dieses war in der Lage, sowohl alle sechs Dimensionen einer Bewegung zu messen als auch Kräfte für alle drei Translationsrichtungen auszugeben (die Ausgabe von Drehkräften wurde nicht unterstützt). Man hoffte, mit diesem Gerät das oben beschriebene Ziel erreichen zu können.

Leider stellte sich bald heraus, dass zwar das Krafrückkopplungsgerät nun ausreichend gut arbeitete, dass aber die zur Verfügung stehenden Computer nicht in der Lage waren, schnell genug komplexere Kraftberechnungen durchzuführen. Die Berechnung von intermolekularen Kräften war nicht zu verwirklichen.

In den folgenden Jahren wurden immer wieder neue GROPE-Systeme entwickelt, aber erst Ende der 1980er Jahre konnte das Molekül-Docking verwirklicht werden. Es wurde ein nochmals leicht verbessertes Krafrückkopplungsgerät und eine wesentlich weiter entwickelte Software eingesetzt. Durch die in der Zwischenzeit entwickelte Computergraphik konnten komplexere Moleküle mit Hilfe des Computers visualisiert und mit einer vereinfachten Berechnung auch die intermolekularen Kräfte ausgegeben werden.

Obwohl trotz einer vereinfachten Berechnung nur 80 Kraft-Updates pro Sekunde berechnet werden konnten, waren die Ergebnisse sehr ermutigend. Die Docking-Resultate konnten mit Kraftausgabe nicht nur schneller erreicht werden, sie waren auch besser als die mit den damals verfügbaren Algorithmen erzielten Ergebnisse.

Aus dem GROPE-Projekt ging später ein weiteres Projekt (der Nano-Manipulator [54]) hervor, das ebenfalls mit Krafrückkopplungsgeräten arbeitet. Ein großes Verdienst des GROPE-Projekts liegt vor allem in der Grundlagenforschung, in der die Relevanz des haptischen Renderns für interaktive Programme durch viele Untersuchungen nachgewiesen wurde.

3.6.2 IMD

Interactive Molecular Dynamics [51] (ab hier kurz IMD) ist ein Projekt der Theoretical Biophysics Group von Prof. Schulten an der Universität von Illinois in Urbana-Champaign/USA. Die Gruppe forscht an Molekular Dynamik Simulationen und hat dazu sowohl das Simulationsprogramm NAMD [30] als auch das Visualisierungsprogramm VMD [25] entwickelt.

Da für die Molekulardynamik-Simulation Kräfte berechnet werden müssen, lag der Gedanke nahe, diese mit Hilfe eines Krafterückkopplungsgerätes auch an den Benutzer auszugeben. VMD besitzt eine Schnittstelle zu NAMD, mit der die aktuellen Simulationsergebnisse zur Laufzeit visualisiert werden können. Diese Schnittstelle wurde erweitert, um auch die Kraftwerte von NAMD an die Visualisierung weiterleiten zu können. Um auch noch ein Krafterückkopplungsgerät zu integrieren, wurde VMD um eine Schnittstelle zu VRPN [55] erweitert. VRPN steuert dabei per Netzwerk einen PC, an den ein Phantom angeschlossen ist.

Nach dem Start aller Programme ist der Benutzer in der Lage, ein Atom mit dem Force-Feedback-Gerät zu steuern. Dabei werden sowohl die auf das Atom einwirkenden Kräfte an den Benutzer ausgegeben, als auch die Kräfte, die der Benutzer per Bewegung seiner Hand auf das Atom ausübt, in die Simulation eingerechnet. Mit Hilfe dieser Methode ist es möglich, zum Beispiel verschiedene Bewegungspfade selber interaktiv nachzuvollziehen und die dabei auftretenden Widerstände zu spüren. Die Autoren geben als Beispielapplikation die Bewegung eines Ions an, welches durch einen Gramacidin-A-Kanal wandern soll, beziehungsweise vom Benutzer durch diesen Kanal geführt werden kann.

Wie die Autoren in [51] schreiben, gibt es zur Zeit zwei wesentliche Haupthindernisse für den Einsatz von IMD in der Praxis. Zunächst einmal ist die Geschwindigkeit der Molekulardynamik-Simulation ein begrenzender Faktor. Da die Krafterückkopplungsgeräte eine sehr hohe Kraft-Update-Rate benötigen, um einen realistischen Krafteindruck zu erzielen (ca. 1 kHz, siehe auch Abschnitt 3.5), muss die MD-Simulation entsprechend schnell ablaufen. Dies ist jedoch bei komplexen Simulationsszenarien oft nicht möglich, da hier einige hunderttausend Teilchen simuliert werden müssen. Daneben bildet das Netzwerk im zur Zeit existierenden Konzept einen Schwachpunkt. Selbst wenn die Kraftberechnung mit einer ausreichenden Frequenz erfolgen kann, müssen die Daten vom Supercomputer zum Haptik-System transportiert werden. Durch die hier auftretenden Verzögerungen wird die taktile Wahrnehmung unter Umständen ebenfalls beeinträchtigt.

Kapitel 4

Interaktiver Einpassprozess

In diesem Kapitel wird das Problem des Abbildens von Molekülstrukturdaten in Dichteverteilungen genauer beschrieben. Es wird aufgezeigt, warum sich dieses Problem überhaupt stellt, und danach wird eine neue interaktive Lösung des Problems entwickelt.

4.1 Motivation

Bruce Alberts beschreibt in seinem Artikel “The Cell as a Collection of Protein Machines” [2], wie sich das Verständnis bezüglich der Abläufe in einer Zelle in den letzten Jahren gewandelt hat. Wurde zunächst davon ausgegangen, dass die biologischen Prozesse in einer Zelle hauptsächlich vom Zufall und den Konzentrationen bestimmter Teilchenarten geprägt sind, so wurde diese einfache Sicht in den letzten Jahren mehr und mehr korrigiert. Es entwickelte sich ein komplexeres Modell, welches von sogenannten “zellulären Maschinen” bestimmt ist. Diese Maschinen sind Mehrkomponenten-Makromoleküle, die komplexe Aufgaben übernehmen. Genau wie die makroskopischen Maschinen der Industriebetriebe führen sie verschiedene Arbeitsschritte aus, um ihre Aufgaben zu bewältigen.

Dieses Bild der Zelle als Fabrik ist neu und revolutioniert nicht nur das Verständnis für die Abläufe in einer Zelle, sondern es betrifft auch alle Wissenschaften, die von diesem Verständnis indirekt abhängen. Zum Beispiel wird die Entwicklung neuer Medikamente entscheidend von diesen Entwicklungen geprägt.

Um mehr über die Struktur und die Funktionsweise dieser Maschinen zu erfahren, werden verschiedene Strukturbestimmungstechniken eingesetzt. Die Methoden müssen jedoch besondere Eigenschaften besitzen, um für die Untersuchung der großen Makromoleküle von Nutzen zu sein.

Eine ideale Strukturbestimmungstechnik würde die Beobachtung einer zellulären Maschine in atomarer Auflösung in ihrer natürlichen Umgebung ermöglichen. Um die Funktions-

weise der Maschine (oder einer anderen biologischen Struktur) ermitteln zu können, müsste es zudem möglich sein, die Maschine in Echtzeit abzubilden, damit so Bewegungen und Abläufe erfasst werden können. Die in der Praxis existierenden Techniken erreichen diese Eigenschaften jedoch nicht. Unter anderem ist eine Beobachtung der Maschine in Echtzeit zur Zeit mit einer Strukturbestimmungstechnik nicht möglich.

Mit Hilfe der Roentgenkristallographie lässt sich die Strukturinformation von Molekülen in atomarer Auflösung erstellen. Vorwiegend mit dieser Methode wurde die Struktur von zirka 18000 biologischen Molekülen ermittelt, die über die Protein-Data-Bank [8] frei verfügbar sind. Leider ist der Präparationsprozess vor dem Einsatz der Roentgenkristallographie sehr aufwendig. Weiterhin lassen sich mit dieser Methode wegen Packungseffekten im Gitter nur bestimmte Konformationen abbilden, und ausserdem wurden nur wenige wirklich große Strukturen ermittelt.

Mit Hilfe von auf der Elektronenmikroskopie basierenden Techniken können große biologische Strukturen mit wesentlich weniger Präparationsaufwand beobachtet werden. Die Auflösung dieser Techniken ist jedoch geringer als die der Roentgenkristallographie und liegt typischerweise zwischen 15 und 25 Ångström. Oft existieren daher niedrigaufgelöste Dichteverteilungen großer biologischen Strukturen, für welche die atomare Struktur nicht bekannt ist.

Jedoch ist oftmals offensichtlich, aus welchen Teilkomponenten sich diese große Struktur zusammensetzt. Da schon von über 18000 biologischen Strukturen der atomare Aufbau ermittelt worden ist, ist die Strukturinformation der Teilkomponenten oft vorhanden. Eine häufig eingesetzte Technik kombiniert die zur Verfügung stehenden verschieden aufgelösten Daten und passt die hochaufgelösten Teilkomponenten in die niedrigaufgelösten Dichteverteilungen ein.

Da die Teilbausteine des Moleküls bekannt sind, stellt sich nur noch die Frage, wo sich diese Teilbausteine in der niedrigaufgelösten Aufnahme befinden. Diese Frage bildet die im folgenden kurz Abbildungsproblem genannte Aufgabenstellung.

Rigid-Body Docking und Flexible Fitting

Das vorgestellte Abbildungsproblem geht davon aus, dass die hochaufgelöste Struktur schon perfekt zum niedrigaufgelösten Volumendatensatz passt. Dies ist aber meist nicht der Fall, da beide Datensätze aus verschiedenen Experimenten stammen und die Struktur deformierbar ist. Die Makromoleküle sind nicht statisch, einzelne Teile der Struktur können ihre Lage relativ zum Gesamtmolekül verändern. Daraus folgt, dass zwei Aufnahmen ein und desselben Moleküls in der Regel unterschiedlich aussehen werden, und dass daher oft die Form der hochaufgelösten Struktur an die niedrigaufgelöste Aufnahme anpaßt werden muss.

Die beschriebene Vorgehensweise, indem die Struktur nicht weiter verändert worden ist, wird "Rigid-Body Docking" genannt. Er ist der einfachere Fall, da das Verändern der

Struktur während des Einpassprozesses eine weitere große Anzahl an Freiheitsgraden schafft.

Wird die Struktur während des Abbildungsvorgangs modifiziert, so wird das Verfahren “Flexible Fitting” genannt. Die Modifikation des Moleküls kann nicht in beliebiger Weise geschehen. Ein großes Makromolekül verändert sich nach gewissen Regeln, die nur aufwendig zu simulieren sind. In interaktiven Programmen wird daher auf das Wissen des Experten gesetzt, der das Makromolekül nur auf gesetzmäßige Weise manipulieren sollte.

In algorithmischen Lösungen werden verschiedene vereinfachende Annahmen gemacht. Wriggers setzt in [67] auf molekulare Mechanik. Er begrenzt die Bewegungen auf wenige “Gelenke”, um die sich Teile des Moleküls drehen dürfen. Dies schränkt die Anzahl der Parameter auf ein sinnvolles Mass ein, stellt jedoch nur eine Näherung der in der Realität vorkommenden Bewegungen dar.

Die vorliegende Arbeit beschäftigt sich wie die meisten anderen Arbeiten mit dem Rigid-Body Docking und versucht nicht, die molekulare Struktur während des Abbildungsvorgangs zu verändern. Jedoch wird in Kapitel 8 noch einmal auf das “Flexible-Fitting” eingegangen und eine mögliche Erweiterung des hier vorgestellten Ansatzes diskutiert.

Interaktive und algorithmische Verfahren

Generell können die existierenden Lösungsmethoden in zwei große Gruppen unterteilt werden. Eine Gruppe von Programmen versucht das Problem interaktiv, eine andere Gruppe algorithmisch zu lösen.

Interaktiv bedeutet in diesem Zusammenhang, dass der Benutzer durch ein Visualisierungsprogramm unterstützt manuell die Strukturdaten in die Dichteverteilung bewegt und so die Abbildung herstellt. Ein Nachteil dieses Verfahrens ist die Komplexität der Aufgabe für den Benutzer. Da das Problem dreidimensionaler Natur ist, hat dieser viele Möglichkeiten, das Molekül zu bewegen und zu drehen. Weiterhin ist die Darstellung am Computerbildschirm oft nicht einfach zu erfassen, und es fällt dem Benutzer schwer, sich in dem dort dargestellten Raum zu orientieren. Zudem sind die klassischen Eingabegeräte wie die Computermaus zweidimensional ausgelegt, so dass weitere Schwierigkeiten bei der Interaktion mit dreidimensionalen Objekten entstehen. Lösungsmöglichkeiten bieten unter anderem die in Kapitel 2 besprochenen Virtual-Reality-Techniken. Ein weiteres Problem stellt die Bewertung der aktuellen Abbildungsposition dar, die der Benutzer rein auf der Grundlage von visuellen Informationen abschätzen muss.

Algorithmisch bedeutet im Idealfall, dass das Problem ohne weitere Eingriffe durch den Benutzer vollautomatisch von einem Computerprogramm gelöst wird. Es existieren hierzu verschiedene Ansätze, die später in Kapitel 5 besprochen werden. Nachteile dieser Algorithmen sind im allgemeinen in der Komplexität des Problems und der damit verbundenen Laufzeit zu sehen.

In Kapitel 5 werden sowohl interaktive als auch algorithmische Ansätze beschrieben und verglichen, sowie die hier vorgeschlagene Lösung eingeordnet.

Im folgenden wird eine neue interaktive Lösung des Problems vorgeschlagen. Diese Lösung vermeidet die Probleme der klassischen interaktiven Programme durch die Anwendung von Virtual-Reality-Techniken. Weiterhin wird haptisches Rendern eingesetzt, um die Komplexität der Aufgabe für den Anwender zu reduzieren.

4.2 Abbildungsproblem

Wie im vorhergehenden Abschnitt beschrieben, soll beim hier zu untersuchenden Rigid-Body Docking ein Molekül in eine Dichteverteilung eingepasst werden. Das Molekül wird durch eine Menge von Atomen $A = \{\mathbf{a}_1, \dots, \mathbf{a}_N\}$ repräsentiert, welche durch dreidimensionale Raumkoordinaten $\mathbf{a}_i = (x_i, y_i, z_i)^T$ definiert sind.

Beim Abbildungsvorgang soll das Molekül in die Dichteverteilung verschoben werden, was zunächst einer Translation $\mathbf{T} = (t_x, t_y, t_z)^T$ entspricht. Da verschiedene Translationen möglich sind, werden die Atomkoordinaten als Funktion angegeben:

$$\mathbf{a}_i(\mathbf{T}) = \mathbf{T} + \mathbf{a}_i$$

Beim hier zu definierenden Abbildungsproblem ist ein geeignetes \mathbf{T} gesucht, wobei die \mathbf{a}_i als gegeben und konstant vorausgesetzt werden. Beim erweiterten Ansatz des Flexible-Fitting sind zusätzlich die Atomkoordinaten \mathbf{a}_i variabel, die initiale Position ist aber ebenfalls gegeben. Gesucht wird hier also nicht nur \mathbf{T} sondern auch die Positionsveränderung der Atome $\Delta\mathbf{a}_i$, wobei $\mathbf{a}_i(\mathbf{T}) = \mathbf{T} + (\mathbf{a}_i + \Delta\mathbf{a}_i)$.

Mit der Translation \mathbf{T} alleine kann die Lösung des Abbildungsproblems nicht vollständig beschrieben werden, da so nur die Position des Moleküls, nicht aber seine Lage beschrieben werden kann. Es muss auch eine Lageveränderung des Moleküls möglich sein, welche durch drei Winkel φ , θ und ϕ beschrieben werden kann.

Soll ein Punkt $\mathbf{P} = (x, y, z)^T$ im Raum relativ zum Ursprung um die z Achse mit dem Winkel ϕ gedreht werden, so ergeben sich folgende neue Koordinaten:

$$\begin{aligned} x' &= x \cos \phi - y \sin \phi \\ y' &= x \sin \phi + y \cos \phi \\ z' &= z \end{aligned}$$

Diese Drehung kann auch mit Hilfe einer Rotationsmatrix dargestellt werden:

$$M_\phi = \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

wobei die neuen Punktkoordinaten durch eine Matrixmultiplikation berechnet werden können:

$$\mathbf{P}' = M_\phi \mathbf{P}$$

Analog lassen sich Rotationsmatrizen für die x und y Achse aufstellen. Eine Gesamtrota-tion um alle drei Achsen wird durch eine sukzessive Drehung erreicht:

$$\mathbf{P}' = M_\varphi M_\theta M_\phi \mathbf{P}$$

Auf diese Weise lassen sich die Atomkoordinaten drehen, so dass sich folgende Gleichung ergibt:

$$\mathbf{a}_i(\mathbf{T}, R) = M_\varphi M_\theta M_\phi \mathbf{a}_i + \mathbf{T} \text{ mit } R = (\varphi, \theta, \phi)$$

Zu beachten ist, dass bei der Drehung um den Koordinatenursprung gedreht wird. Das Molekül soll jedoch als Körper seine Orientierung verändern, was einer Drehung um den Mittelpunkt \mathbf{O} entspricht. Dies kann erreicht werden, indem vor der Drehung das Molekül in den Ursprung verschoben und es nach der Drehung wieder an seine ursprüngliche Position versetzt wird:

$$\mathbf{a}_i(\mathbf{T}, R) = M_\varphi M_\theta M_\phi (\mathbf{a}_i - \mathbf{O}) + \mathbf{O} + \mathbf{T}$$

Das Molekül soll wie oben beschrieben in eine Dichteverteilung abgebildet werden. Die Dichteverteilung ist mathematisch ein dreidimensionales Skalarfeld $\rho_{em}(x, y, z)$, bei dem an jeder Position x, y, z mit $x \in \{0, \dots, x_{max}\}$, $y \in \{0, \dots, y_{max}\}$ und $z \in \{0, \dots, z_{max}\}$ ein skalarer Wert $\rho_{em}(x, y, z) \in \mathbb{R}$ vorhanden ist. Physikalisch ist die Dichteverteilung das Resultat einer Mikroskopierekonstruktion, welche ein dreidimensionales Gebiet im Raum abbildet. Die Dichteverteilung besitzt also eine räumliche Ausdehnung, das heisst der Wert $\rho_{em}(x, y, z)$ wird nicht nur einem Punkt im Raum, sondern einem Voxel (entspricht einem Würfel) zugeordnet. Innerhalb dieses Voxels besitzen alle möglichen Raumpositionen immer den gleichen Wert.

Wird zu einer beliebigen Raumposition der Wert des Voxels gesucht, so kann der nächste benachbarte Skalarfeldeintrag ermittelt werden. Dies führt jedoch zu sprunghaften Wertänderungen an den Voxelgrenzen, was durch den Einsatz von Interpolation vermieden werden kann. Interpolationsfunktionen werden später in diesem Kapitel vorgestellt.

Soll diese Abbildung des Moleküls in die Dichteverteilung vorgenommen werden, so sollte das Ziel sein, eine möglichst optimale Abbildungsposition und -orientierung zu finden. Um eine solche optimale Position finden zu können, muss ein Maß vorhanden sein, mit dessen Hilfe verschiedene Positionen und Orientierungen bewertet und verglichen werden können. Dieses Maß entspricht allgemein einer Funktion C .

Mit Hilfe dieser Komponenten kann das Abbildungsproblem wie folgt definiert werden:

Definition 4.2.1 Abbildungsproblem

Gegeben sei ein Molekül in Form einer Menge von Koordinaten $A = \{\mathbf{a}_1, \dots, \mathbf{a}_N\}$ mit $\mathbf{a}_i \in \mathbb{R}^3$. Weiterhin sei eine Dichteverteilung in Form eines Skalarfeldes $\rho_{em}(x, y, z)$ mit $x \in \{0, \dots, x_{max}\}$, $y \in \{0, \dots, y_{max}\}$ und $z \in \{0, \dots, z_{max}\}$ und $\rho_{em}(x, y, z) \in \mathbb{R}$ gegeben. Jeder Wert $\rho_{em}(x, y, z)$ entspreche einem dreidimensionalen Voxel mit der Kantenlänge σ_V .

Eine Lösung des “Rigid-Body Docking”-Problems für eine gegebene Funktion $C(\rho_{em}, A)$ stellt eine Position $\mathbf{T} = (t_x, t_y, t_z)^T$ und eine Rotation $R = (\varphi, \theta, \phi)$ dar, für die die Funktion $C(\rho_{em}, A(\mathbf{T}, R))$ maximal wird. Hierbei drückt $A(\mathbf{T}, R)$ aus, dass die Koordinaten A an die Position \mathbf{T} verschoben und um den Mittelpunkt \mathbf{O} von A mit den Winkeln $R = (\varphi, \theta, \phi)$ rotiert werden. \diamond

Im folgenden wird die Funktion C zur Vereinfachung der Schreibweise oft nur mit den zwei gesuchten Parametern \mathbf{T} und R als $C(\mathbf{T}, R)$ dargestellt und die beiden Parameter ρ_{em} und A , da sie im Verlauf des Verfahrens konstant sind, weggelassen.

Ein algorithmisches Programm sucht vollautomatisch innerhalb des Suchraums nach einem globalen Maximum von $C(\mathbf{T}, R)$. Der Suchraum ist sehr groß, da $\mathbf{T} \in \mathbb{R}^3$ mit $0 \leq t_x \leq \sigma_V x_{max}$, $0 \leq t_y \leq \sigma_V y_{max}$, $0 \leq t_z \leq \sigma_V z_{max}$, sowie $R \in \mathbb{R}^3$ mit $0 \leq \varphi, \theta, \phi < 2\pi$. Zu beachten ist, dass \mathbf{T} und R voneinander unabhängige Größen sind, es existieren also für jede mögliche Position \mathbf{T} alle verschiedenen Orientierungen R .

Da sowohl \mathbf{T} als auch R reellwertig sind, ist eine vollständige Untersuchung des gesamten Suchraums nicht möglich. Es wird meist eine konstante Schrittweite eingesetzt, zum Beispiel $t_x = ns_x$ mit $n = 0, \dots, \left\lfloor \frac{\sigma_V x_{max}}{s_x} \right\rfloor$. So wird die zuvor unendliche Menge an Möglichkeiten auf eine endliche Menge reduziert.

4.3 Entwicklung einer Bewertungsfunktion

Für das beschriebene Abbildungsproblem wird im folgenden eine geeignete Funktion C entwickelt. Diese Funktion soll eine konkrete Abbildungssituation des Moleküls in die Dichteverteilung bewerten. Wie zu Beginn schon beschrieben, stellt die Molekülstruktur ein dreidimensionales Objekt dar, welches in der Dichteverteilung “wiedererkannt” werden soll.

Ähnliche Problemstellungen treten in vielen anderen Forschungsgebieten ebenfalls auf, zum Beispiel in der Bildverarbeitung oder auch der Spracherkennung. Einen allgemeinen Ansatz zur Lösung stellt die Signaltheorie zur Verfügung. Hier werden die zu vergleichenden Datensätze allgemein als Signale interpretiert.

Um die signaltheoretischen Methoden anwenden zu können, müssen die beiden Datensätze als vergleichbare Signale vorliegen. Zunächst wurde die Molekülstruktur als Menge von

Atompositionen beschrieben. Da jedoch jedes Atom nicht nur eine Raumposition sondern auch eine Ausdehnung besitzt, liegt es nahe, die Molekülstruktur ebenfalls als dreidimensionale Dichteverteilung darzustellen.

Um aus den punktförmigen Koordinaten einen dreidimensionalen Volumendatensatz zu erzeugen, können verschiedene Ansätze verfolgt werden. In dieser Arbeit wurde der Ansatz von Wriggers [64] verwendet. Er setzt einen Gaussfilter ein, der ebenfalls in der Bildverarbeitung angewendet wird.

4.3.1 Filter in der Bildverarbeitung

Bilder weisen oft verschiedene Defekte auf, die durch Bildbearbeitungsmethoden korrigiert werden können. Als Beispiel sei hier Rauschen genannt.

Hochfrequentes Rauschen äussert sich in der Praxis in vereinzelt Pixeln, die im Gegensatz zu ihren Nachbarn einen stark unterschiedlichen Helligkeits- oder Farbwert besitzen. Die einfachste Möglichkeit diesen Defekt zu beheben, besteht in der Mittelung zwischen benachbarten Pixeln. Im einfachsten Fall wird der Mittelwert der Farbwerte jedes einzelnen Pixels und seiner acht Nachbarn berechnet:

$$p^*(x, y) = \frac{1}{9} \sum_{i,j=-1}^1 p(x+i, y+j)$$

Bei der Betrachtung des Resultats wird dann meist festgestellt, dass das Rauschen noch nicht vollständig verschwunden ist. Nun könnte für jedes Pixel ein größerer Bereich zur Mittelwertbildung herangezogen werden. Aber auch hierbei sind die Resultate oft nicht zufriedenstellend. Das Rauschen verschwindet zwar immer stärker, ebenso aber auch feine Details, die durch die Mittelwertbildung immer weiter "verwischt" werden.

Ein verallgemeinerter Ansatz macht nicht nur die Größe des zur Mittelwertbildung herangezogenen Bereichs variabel, sondern gewichtet die einzelnen Pixel auch unterschiedlich:

$$p^*(x, y) = \frac{\sum_{i,j=-m}^m w(i, j) p(x+i, y+j)}{\sum_{i,j=-m}^m w(i, j)}$$

wobei $p(x, y)$ die Pixel und $w(i, j)$ die Gewichte repräsentieren. Es ergibt sich ein $(2m+1) \times (2m+1)$ großer Bereich, der für die Filterung herangezogen wird. Das oben beschriebene Szenario der Mittelwertbildung mit allen direkten Nachbarn würde hier $m = 1$ und folgender Gewichtsmatrix $w(i, j)$ entsprechen:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



Abbildung 4.1: Mittelwert-Filter

Die Gewichtsmatrix wird auch Kernel genannt. In Abbildung 4.1 wurde auf ein Testbild (links) ein solcher Kernel (jedoch mit $m = 2$) angewendet.

Neben der reinen Mittelwertbildung sind Kernel nützlich, die den ursprünglichen Pixelwert stärker betonen und die Nachbarn schwächer gewichten:

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Für das Aufstellen geeigneter Kernel werden meist Funktionen verwendet, zum Beispiel die Gauss-Funktion:

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma_G} e^{-\frac{x^2}{2\sigma_G^2}}$$

welche der Dichtefunktion der Normalverteilung entspricht. Aus dieser eindimensionalen Funktion können durch Multiplikation auch leicht mehrdimensionale Formen hergeleitet werden:

$$\begin{aligned} g(x, y) &= g(x) g(y) \\ &= \frac{1}{2\pi\sigma_G^2} e^{-\frac{x^2+y^2}{2\sigma_G^2}} \end{aligned}$$

Zum Aufstellen eines Kernels wird die kontinuierliche Funktion auf ein diskretes Gitter abgebildet, was zum Beispiel zu folgendem Kernel führt:

$$\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

Ein Vorteil des Gaussfilters ist, dass er hohe Frequenzen (zum Beispiel Rauschen) ausreichend dämpft und trotzdem schnell, aber nicht abrupt, abfällt. Der Parameter σ_G entspricht

der Standardabweichung der Normalverteilung, so dass anschaulich ein steigendes σ_G zu einem flacheren Auslaufen der Funktion führt. Der Einfluss des ursprünglichen Pixelwertes wird so schwächer, der Filter verwischt die Informationen stärker.

Nachteil der vorgestellten Kerneloperationen ist der Rechenzeitbedarf. Bei größerem Kernel steigt die Anzahl an Operationen pro Pixel an, was bei großen Bildern zu einer langen Rechenzeit führt. Es kann die Anwendung eines solchen Kernels aber auch als Faltung interpretiert werden:

$$p^*(x, y) = g(x, y) * p(x, y) = \int \int (p(\alpha, \beta) g(x - \alpha, y - \beta) d\alpha d\beta$$

Die Faltungsoperation entspricht im Frequenzraum einer einfachen Multiplikation, also:

$$g(x, y) * p(x, y) \Leftrightarrow G(u, v) P(u, v)$$

wobei $G(u, v)$ und $P(u, v)$ die Fouriertransformierten von $g(x, y)$ und $p(x, y)$ seien. Bei großen Bildern und komplexen Kernen bietet es sich also an, die Daten zunächst per Fouriertransformation in den Frequenzraum zu transformieren, um sie dort leichter bearbeiten zu können. Danach ist eine Rücktransformation zu vollziehen.

4.3.2 Erzeugen einer Dichteverteilung

Die im vorhergehenden Abschnitt beschriebenen Methoden können für die Erzeugung einer Dichteverteilung aus einer Molekülstruktur herangezogen werden. Die Grundidee ist, dass statt zweidimensionalen Bilddaten dreidimensionale Volumendaten mit einem Filter bearbeitet werden.

Wriggers [64] bildet dazu die Atomkoordinaten zunächst mit Hilfe trilinearer Interpolation auf das Gitter eines Volumendatensatzes ρ_A ab. Ein einzelnes Atom a_i trägt dabei zu acht Voxelwerten bei, wobei jedes Voxel eine Breite von σ_V besitzt. Es sei $x = \left\lfloor \frac{a_{ix}}{\sigma_V} \right\rfloor$, $y = \left\lfloor \frac{a_{iy}}{\sigma_V} \right\rfloor$, $z = \left\lfloor \frac{a_{iz}}{\sigma_V} \right\rfloor$ und $\Delta\rho_A(x, y, z)$ sei der Beitrag des Atoms i zu dem Voxelwert $\rho_A(x, y, z)$.

$$\Delta\rho_A(x, y, z) = \left(1 - \frac{a_{ix} - x\sigma_V}{\sigma_V}\right) \left(1 - \frac{a_{iy} - y\sigma_V}{\sigma_V}\right) \left(1 - \frac{a_{iz} - z\sigma_V}{\sigma_V}\right) m_i$$

$$\Delta\rho_A(x + 1, y, z) = \left(\frac{a_{ix} - x\sigma_V}{\sigma_V}\right) \left(1 - \frac{a_{iy} - y\sigma_V}{\sigma_V}\right) \left(1 - \frac{a_{iz} - z\sigma_V}{\sigma_V}\right) m_i$$

Die Masse eines einzelnen Atoms geht über den Faktor m_i in die Gleichung ein. Der Beitrag des Atoms zu den anderen Gitterpunkten ergibt sich analog.

Auf diesen Volumendatensatz werden danach Filter ähnlich den oben beschriebenen angewendet. Dadurch werden die einzelnen Voxel dreidimensional “verwischt”, ganz analog zu dem Effekt, der bei den zweidimensionalen Bildern auftritt. Ist σ_V groß, so verlaufen die Informationen der einzelnen Atome ineinander. Dies entspricht genau der Eigenschaft, die auch bei schwach aufgelösten Mikroskopieaufnahmen festgestellt werden kann. Durch die Intensität der Filteroperation (beim Gauss-Filter über die Standardabweichung σ_G einstellbar) kann so die Auflösung der simulierten Dichteverteilungen eingestellt werden, wobei hier $\sigma_G \gg \sigma_V$ gelten soll. Durch die Anwendung des Filters entsteht ein neues dreidimensionales Skalarfeld ρ_{calc} :

$$\rho_{calc}(x, y, z) = \rho_A(x, y, z) * g(x, y, z)$$

Wird ein dreidimensionaler Gauss-Filter auf ein Molekül mit steigendem σ_G (also mit immer weiter sinkender Auflösung) angewendet, so entstehen die Volumen in Abbildung 4.2.

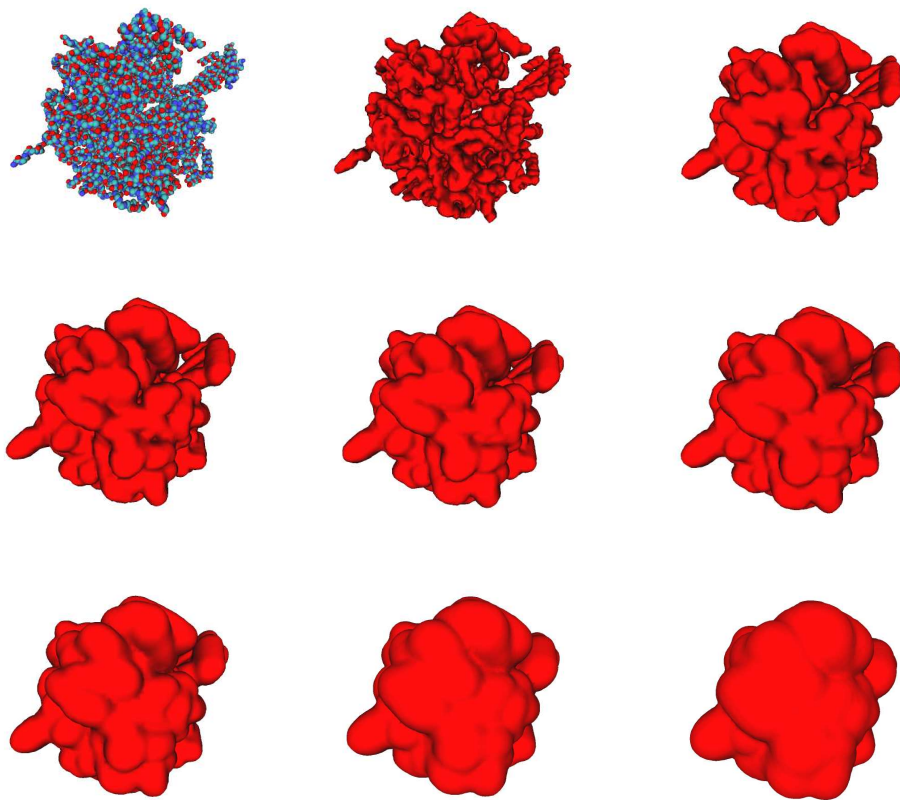


Abbildung 4.2: Berechnete Dichteverteilungen

Einen zu dieser Methode ähnlichen Ansatz findet man in [6].

Durch die hier vorgestellte Methode kann aus einer Molekülstruktur ein dreidimensionaler Volumendatensatz ρ_{calc} mit einstellbarer Auflösung erzeugt werden. Auf diese Weise entsteht so ein vergleichbarer Datensatz zur experimentell ermittelten Dichteverteilung ρ_{em} .

4.3.3 Korrelation

Die mit der im vorherigen Abschnitt beschriebenen Methode erzeugte Dichteverteilung ρ_{calc} und die experimentell ermittelte Dichteverteilung ρ_{em} sollen miteinander verglichen werden. Eine konkrete Abbildung kann über die Parameter \mathbf{T} für den translativen Anteil und den Parameter R für rotationalen Anteil spezifiziert werden.

Zur Bewertung wird die quadratische Abweichung der beiden Datensätze verwendet:

$$D(R, \mathbf{T}) = \int \int \int (\rho_{calc}(R, \mathbf{T}, x, y, z) - \rho_{em}(x, y, z))^2 dx dy dz$$

Um eine optimale Abbildungsposition zu finden, muss diese Funktion D abhängig von R und \mathbf{T} minimiert werden. Die Funktion entspricht:

$$\begin{aligned} D(R, \mathbf{T}) &= \underbrace{\int \int \int (\rho_{calc}(R, \mathbf{T}, x, y, z))^2 dx dy dz}_{\text{konstant}} \\ &+ \underbrace{\int \int \int (\rho_{em}(x, y, z))^2 dx dy dz}_{\text{konstant}} \\ &- 2 \underbrace{\int \int \int \rho_{calc}(R, \mathbf{T}, x, y, z) \rho_{em}(x, y, z) dx dy dz}_{C(R, \mathbf{T})} \end{aligned}$$

Um die Abweichung zu minimieren muss der letzte Term $C(R, \mathbf{T})$ maximiert werden, die beiden konstanten Terme können für die Minimierung vernachlässigt werden. Die Funktion $C(R, \mathbf{T})$ wird auch als Korrelationsfunktion bezeichnet:

Definition 4.3.1 Korrelation

Die Funktion

$$C(R, \mathbf{T}) = \int \int \int \rho_{calc}(R, \mathbf{T}, x, y, z) \rho_{em}(x, y, z) dx dy dz$$

wird als *Korrelation* der beiden Dichteverteilungen ρ_{calc} und ρ_{em} für eine Abbildung R und \mathbf{T} bezeichnet. \diamond

Die Korrelationsfunktion kann auch mit Hilfe der stochastischen Signaltheorie [20] hergeleitet werden.

4.3.4 Interpolation

Im vorliegenden Fall sollen die gerade vorgestellten Korrelationsfunktionen für das Abbilden von zwei Volumendatensätzen genutzt werden. Da die Volumendatensätze diskrete Daten darstellen, müssen zunächst aus den oben angeführten Funktionen diskrete Formen hergeleitet werden. Die beiden Datensätze ρ_{calc} und ρ_{em} sind im diskreten Fall dreidimensionale Skalarfelder, so dass sich für die Korrelation folgende diskrete Form ergibt:

$$C(R, \mathbf{T}) = \sum_{x=1}^N \sum_{y=1}^N \sum_{z=1}^N \rho_{calc}(R, \mathbf{T}, x, y, z) \rho_{em}(x, y, z)$$

Bei der Suche nach einer maximalen Korrelation, also nach der besten Abbildungsposition, müssen verschiedene mögliche Abbildungen untersucht werden. Wird davon ausgegangen, dass hierbei nur der Datensatz ρ_{calc} in der Position verändert wird, so ergibt sich:

$$C(R, \mathbf{T}) = \sum_{x=1}^N \sum_{y=1}^N \sum_{z=1}^N \rho_{calc}(R, x + t_x, y + t_y, z + t_z) \rho_{em}(x, y, z) \text{ mit } \mathbf{T} = (t_x, t_y, t_z)^T$$

Die obige Formel lässt sich jedoch nur dann unverändert einsetzen, wenn die Komponenten von \mathbf{T} immer ganzzahlig sind. Für nicht ganzzahlige t_x, t_y, t_z liegt kein Wert des Skalarfeldes (also der Dichteverteilung) vor. Soll jedoch zum Beispiel die Korrelation bei einer Verschiebung um $\frac{1}{3}$ Voxelbreite bestimmt werden, so ist dies mit obiger Funktion nicht möglich. Noch problematischer wird die Anwendung der Formel, wenn die dreidimensionalen Volumendaten auch gedreht abgebildet werden sollen.

Allgemein bedeutet dieses Problem, dass Werte eines Skalarfeldes an Stellen berechnet werden sollen, an denen keine Werte vorliegen. Analoge Probleme treten auch in der Bildverarbeitung auf, wenn Bilder transformiert werden sollen — zum Beispiel eine Drehung $\neq 90^\circ$. Allgemein bedeutet dies, dass Funktionswerte $f(x, y)$ für Positionen (x, y) gesucht werden, die jedoch nicht mit Pixelpositionen (m, n) übereinstimmen. Diese Funktionswerte $f(x, y)$ müssen aus den vorhandenen Pixelwerten $f(m, n)$ berechnet werden. Dies kann als Faltungsoperation mit einem Kernel $h(x, y)$ dargestellt werden, analog zu den in Abschnitt 4.3.1 vorgestellten Filteroperationen:

$$f(x, y) = \sum_m \sum_n f(m, n) h(x - m, y - n)$$

Zur Vereinfachung wird für den Kernel die Separierbarkeit angenommen, also $h(x, y) = h_s(x)h_s(y)$.

Interpolation 0. Ordnung

Eine sehr naheliegende Möglichkeit, den fehlenden Funktionswert $f(x, y)$ zu berechnen, ist die Suche nach dem nächsten Nachbarpixel $f(m, n)$. Dies entspricht einer Faltung mit dem Kernel

$$h_0(x) = \begin{cases} 1 & 0 \leq |x| < 0,5 \\ 0 & \text{sonst} \end{cases}$$

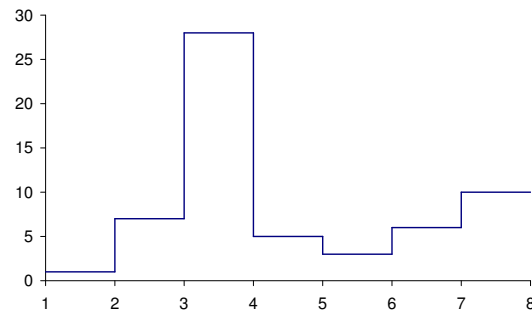


Abbildung 4.3: Interpolation 0. Ordnung

Jeder Pixel bestimmt so einen Bereich der Funktion $f(x, y)$. Der entstehende Funktionsverlauf von $f(x, y)$ (ein Beispiel ist dargestellt in Abbildung 4.3) besitzt an den Übergängen der Bereiche Sprungstellen zum nächsten Pixelwert. Es ist meist sinnvoller, diese harten Übergänge zu vermeiden.

Interpolation 1. Ordnung

Bei der Interpolation erster Ordnung werden die Werte zwischen zwei Pixeln linear aus den vorhandenen Pixelwerten berechnet. Dies entspricht einer Faltung mit dem Kernel

$$h_1(x) = \begin{cases} 1 - |x| & 0 \leq |x| < 1 \\ 0 & \text{sonst} \end{cases}$$

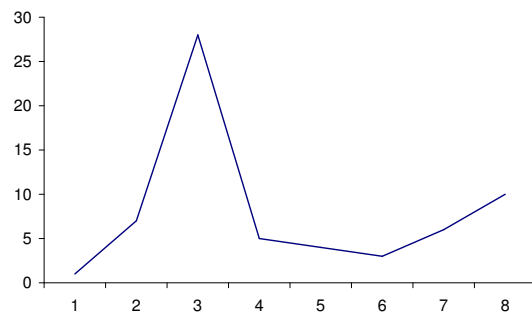


Abbildung 4.4: Interpolation 1. Ordnung

Diese lineare Interpolation (Beispiel in Abbildung 4.4) wird aufgrund des geringen Rechenaufwandes oft angewendet.

Interpolation n-ter Ordnung

Der Kernel erster Ordnung kann auch als Faltung zweier Kernel der Interpolation nullter Ordnung dargestellt werden, also

$$h_1(x) = h_0(x) * h_0(x)$$

Allgemein kann so die Interpolation n-ter Ordnung definiert werden:

$$h_n(x) = \underbrace{h_0(x) * h_0(x) * \dots * h_0(x)}_{n \text{ mal}}$$

Für $n \rightarrow \infty$ nimmt h_n die Gestalt des Gauss-Kernels an. Der Rechenaufwand steigt für größere n deutlich an, so daß im konkreten Anwendungsfall überprüft werden muss, ob die erreichbare Verbesserung des Ergebnisses dies auch rechtfertigt.

Anwendung der Interpolation

Wird für das vorliegende Problem der Rotation und Translation der diskreten Volumendatensätze eine allgemeine Interpolationsfunktion I eingesetzt, so ergibt sich folgende Korrelationsfunktion:

$$C(R, \mathbf{T}) = \sum_{x=1}^N \sum_{y=1}^N \sum_{z=1}^N I(\rho_{calc}, x, y, z, R, \mathbf{T}) \rho_{em}(x, y, z)$$

Die Interpolationsfunktion $I(\rho_{calc}, x, y, z, R, \mathbf{T})$ berechnet für einen diskreten Volumendatensatz ρ_{calc} und für eine Rotation R und Translation \mathbf{T} den Volumenwert an der Stelle x, y, z .

4.4 Einfaches Verfahren

Nachdem nun die notwendigen Werkzeuge vorgestellt worden sind, kann mit deren Hilfe ein erster Verfahrensvorschlag entworfen werden.

Es soll eine Molekülstruktur in eine Dichteverteilung abgebildet werden. Hierzu wird zunächst per Anwendung eines Filters, wie in Abschnitt 4.3.2 beschrieben, ein Volumendatensatz aus den Strukturdaten berechnet. Da das Molekül beziehungsweise das berechnete Volumen verschiedene Abbildungspositionen einnehmen kann, müssen diese Positionen

miteinander verglichen werden. Hierzu wird eine der in Abschnitt 4.3.3 vorgestellten Korrelationsfunktionen als Bewertungsfunktion eingesetzt. Wird der berechnete Volumendatensatz verschoben oder gedreht, so muss ein neuer Volumendatensatz berechnet werden. Hierzu können die in Abschnitt 4.3.4 vorgestellten Interpolationsmethoden eingesetzt werden.

Mit Hilfe dieser Methoden können verschiedene Abbildungssituationen hergestellt und verglichen werden. In der vorliegenden Arbeit soll der Benutzer interaktiv die Position und Orientierung des Moleküls ändern können. Dabei werden die beiden Volumendatensätze graphisch dreidimensional dargestellt. Um diese dreidimensionale Darstellung möglichst realistisch gestalten zu können, werden hier Virtual-Reality-Methoden eingesetzt, die schon in Kapitel 2.2 vorgestellt worden sind. Sie sind in der Lage, den Tiefeneindruck durch eine stereoskopische Darstellung wesentlich zu steigern und können so eine Orientierung des Benutzers im virtuellen Raum erleichtern.

Im Rahmen dieses interaktiven Einpassprozesses soll der Benutzer von haptischem Rendern unterstützt werden. Die Grundidee der Arbeit ist, dass das haptische Rendern als weitere Informationsquelle, zusätzlich zur visuellen Information, eingesetzt wird. Der Benutzer soll hierbei eine Kraft verspüren, die ihn in Richtung einer besseren Abbildungsposition zieht. Besser bedeutet hier, dass die Bewertungsfunktion C in Richtung der Kraft ansteigt. Folgt der Benutzer dieser Kraft, so erreicht er ein lokales Maximum von C . Die Kraft soll jedoch nur so schwach sein, dass der Benutzer weiterhin die Möglichkeit hat, ihr zu widerstehen und so selbstständig andere Abbildungspositionen ansteuern kann. Er ist daher verantwortlich für die Suche nach dem globalen Maximum. Da er aber als Experte über ein enormes Kontextwissen über den Datensatz verfügt, ist er hier gegenüber algorithmischen Lösungen im Vorteil.

Das Verfahren soll von Forschern aus dem Bereich der strukturellen Biologie eingesetzt werden. Zu beachten ist, dass diese in der Regel nicht über große Supercomputer verfügen. Daher soll die Lösung auch auf normalen Standardworkstations einsatzfähig sein. Im folgenden wird ein Verfahren beschrieben, welches diesen Vorschlag umsetzt.

Verfahren

Zunächst müssen die beiden abzubildenden Datensätze eingeladen werden. Im vorliegenden Fall wird eine Molekülstruktur in Form einer PDB Datei [9, 8], sowie eine Dichteverteilung in Form einer SITUS Datei [68] eingelesen. Die Dichteverteilung stellt einen Volumendatensatz dar, der zunächst nicht direkt mit der Strukturdatei verglichen werden kann. Also wird in einem zweiten Schritt das Molekül in der Auflösung an die Dichteverteilung angepasst und in einen Volumendatensatz transformiert.

Nach dieser Initialisierungsphase (Abbildung 4.5) wird der eigentliche Einpassprozess gestartet. Der Einpassprozess soll interaktiv, mit Hilfe von haptischem Rendern erfolgen.

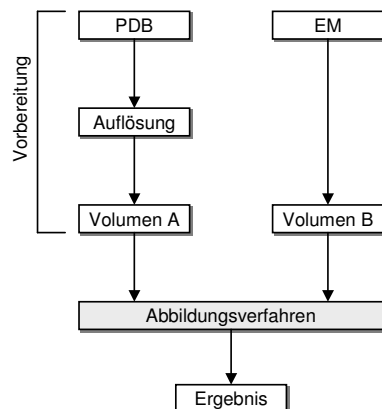


Abbildung 4.5: Initialisierungsphase

Hierzu soll die Molekülstruktur mit Hilfe eines haptischen Gerätes in die Dichteverteilung bewegt werden. Das haptische Rendern soll den Anwender in diesem Prozess unterstützen, indem das Gerät ihn in die Richtung einer besseren Abbildungsposition zieht. Um dies zu erreichen, muss die aktuelle Position mit Hilfe der Korrelationskoeffizienten bewertet und hieraus eine geeignete Kraft berechnet werden.

Hierzu wird in einem ersten Schritt die aktuelle Position und Orientierung des haptischen Gerätes ermittelt. Diese Position und Orientierung wird auf den ersten Volumendatensatz angewendet (die in der Auflösung angepasste Molekülstruktur). In einem nächsten Schritt muss aus dem gedrehten Volumen wieder ein zum statischen Volumen passender Datensatz per Interpolation berechnet werden. Ebenso muss die graphische Darstellung neu gezeichnet werden.

Stehen zwei zueinander passende Volumendatensätze zur Verfügung, so kann die Korrelation der beiden berechnet werden. Aus dieser aktuellen Korrelation wird danach eine geeignete Kraft für das haptische Rendern berechnet und sogleich mit Hilfe des Phantoms an den Benutzer ausgegeben.

Diese Prozedur wird so lange wiederholt, bis der Anwender mit der aktuellen Abbildungsposition zufrieden ist. Dies wird vor allem dann der Fall sein, wenn die ausgegebene Kraft sehr gering ist und der Benutzer zudem mit Hilfe des ihm zur Verfügung stehenden Kontextwissens die Situation positiv bewertet hat.

Der gesamte Einpassprozess ist in Abbildung 4.6 dargestellt.

Dieser interaktive Lösungsvorschlag orientiert sich in seinen Berechnungen an bereits bewährten algorithmischen Programmen, welche im folgenden Kapitel vorgestellt werden.

Erste Untersuchungen und Erfahrungen bezüglich der Laufzeit algorithmischer Programme zeigten jedoch schnell, dass ein solcher Ansatz nicht mit der notwendigen Geschwindigkeit ablaufen kann. Vor allem die aufwendigen Berechnungen in der Hauptschleife, welche mehrfach die kompletten Volumendatensätze einbeziehen, lassen ein solches Vorhaben

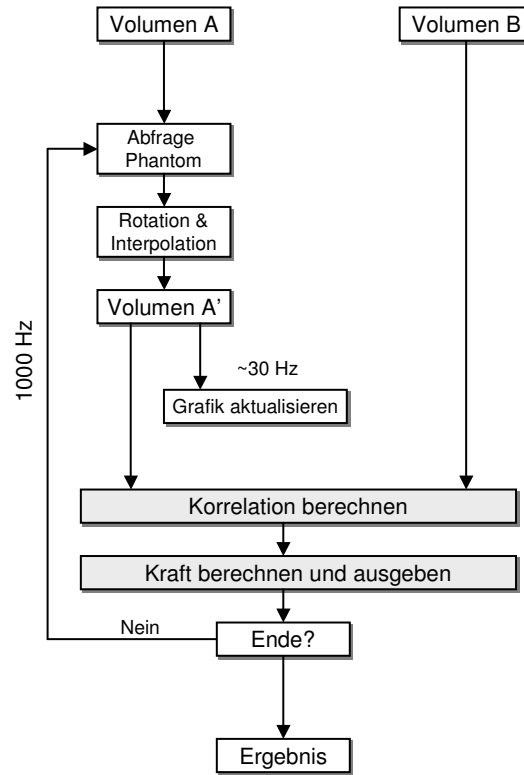


Abbildung 4.6: Einpassprozess

scheitern. Ausserdem stellt das haptische Rendern einige Ansprüche an die Berechnungsgeschwindigkeit. Wie in Kapitel 3.5 beschrieben, sollten in der Regel ungefähr 1000 Kraftberechnungen pro Sekunde erfolgen. Eine solche Frequenz kann aber eine solche komplizierte Berechnungsvorschrift beim Einsatz von heutigen Standard-Workstations nicht erreichen. Vor allem ist die Berechnungsgeschwindigkeit von der Größe der Volumendatensätze abhängig, was für viele interessante Anwendungsgebiete nicht wünschenswert ist.

Vektorquantisierung In dieser Arbeit wird zur Beschleunigung der Kraftberechnung die Methode der Vektorquantisierung eingesetzt. Sie erlaubt es, ein komplexes Signal mit Hilfe von relativ wenigen Merkmalsvektoren zu beschreiben. Je weniger Vektoren zur Beschreibung eingesetzt werden, desto größer ist die Kompression. Da in der Regel nur wenige Vektoren ein komplexes Signal beschreiben, stellt dies eine verlustbehaftete Kompression dar — es kann zumeist das ursprüngliche Signal nicht mehr perfekt aus den Vektoren rekonstruiert werden.

Ein Vektorquantisierer Q und Größe N bildet einen k -dimensionalen Raum auf eine endliche Menge C ab. Diese Menge C bestehe aus N k -dimensionalen Ausgabevektoren \mathbf{w}_i (auch Merkmals- oder Codebookvektoren genannt).

$$Q : \mathbb{R}^k \rightarrow C \text{ wobei } C = \{\mathbf{w}_1, \dots, \mathbf{w}_N\}, \mathbf{w}_i \in \mathbb{R}^k \forall i \in \{1, \dots, N\}$$

Ein Vektor $\mathbf{x} = (x_1, \dots, x_k)^T$ wird auf denjenigen Codebookvektor \mathbf{w}_i abgebildet, zu dem er den geringsten Abstand $\|\mathbf{x} - \mathbf{w}_i\|$ hat:

$$Q(\mathbf{x}) = \mathbf{w}_i \in C : \|\mathbf{x} - \mathbf{w}_i\| \leq \|\mathbf{x} - \mathbf{w}_j\| \quad \forall j \in \{1, \dots, N\}$$

Eingesetzt wird die Vektorquantisierung auf einer konkreten Menge $R = \{\mathbf{r}_1, \dots, \mathbf{r}_M\}$, $r_i \in \mathbb{R}^k \forall i \in \{1, \dots, M\}$, deren Vektoren mit Hilfe der Funktion $Q(\mathbf{r}_i)$ auf die Codebookvektoren abgebildet werden. Wird der ursprüngliche Vektor r_i in späteren Berechnungen durch den zugehörigen Codebookvektor \mathbf{w}_j ersetzt, so entsteht ein Fehler. Dieser Fehler kann zum Beispiel über die Standardabweichung bestimmt werden:

$$E(R, C) = \sqrt{\frac{\sum_{i=1}^N m_i \|\mathbf{r}_i - Q(\mathbf{r}_i)\|^2}{\sum_{i=1}^N m_i}}$$

wobei hier die einzelnen Vektoren aus R mit den Faktoren m_i unterschiedlich gewichtet werden.

Aufgabe einer Vektorquantisierungsmethode ist es, bei einer vorgegebenen Anzahl an Codebookvektoren \mathbf{w}_i diesen Fehler möglichst gering zu halten. Es soll das Signal R mit Hilfe der wenigen Vektoren also möglichst optimal wiedergegeben werden.

Um eine Menge der Codebookvektoren C bezüglich einer Fehlerfunktion E optimal aufzustellen, sind verschiedene Algorithmen entwickelt worden, zum Beispiel der Algorithmus von Linde, Buzo und Gray [34]. Die Aufgabe ist jedoch nicht trivial. Viele Algorithmen laufen Gefahr, nur lokale Minima der Fehlerfunktion zu finden (zum Beispiel der erwähnte LBG Algorithmus), oder sie benötigen sehr viel Rechenzeit.

Im nächsten Abschnitt wird der Vektorquantisierungsalgorithmus der “Topology Representing Networks” von Martinetz und Schulten [36] beschrieben und danach auf das vorliegende Problem angewandt. Hieraus wird ein neuer Lösungsvorschlag entwickelt, welcher sich aufgrund seiner Geschwindigkeit wesentlich besser für das haptische Rendern eignet.

4.5 Topology Representing Networks

Die Topology Representing Networks sind ein Verfahren aus dem Forschungsgebiet der neuronalen Netzwerke. Daher wird im folgenden kurz auf dieses Gebiet eingegangen und danach das eigentliche Verfahren vorgestellt.

4.5.1 Neuronale Netzwerke

Es existieren sowohl künstliche als auch natürliche neuronale Netzwerke. Unser Gehirn stellt ein natürliches neuronales Netzwerk dar, welches aus einer sehr großen Anzahl von neuronalen Zellen (kurz Neuronen) besteht.

Abbildung 4.7 zeigt eine vereinfachte Darstellung einer solchen natürlichen neuronalen Zelle. Ein baumartiges Netzwerk von feinen Nervenfasern, den sogenannten Dendriten, ist mit dem eigentlichen Zellkörper des Neurons verbunden. Von diesem Zellkörper geht eine einzelne dickere Nervenfaser — das Axon — aus, das sich wieder in feinere Fasern aufteilt. An deren Enden sitzen Synapsen, die wiederum mit Dendriten anderer Zellen oder mit anderen Zellkörpern direkt verbunden sind.

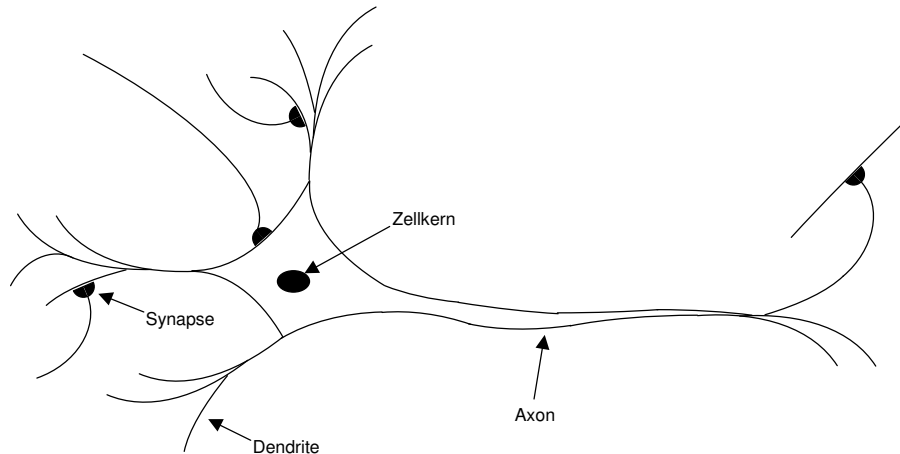


Abbildung 4.7: Natürliches neuronales Netzwerk

Vereinfacht dargestellt kennt ein Neuron nur zwei Zustände: “feuern” oder “nicht feuern”. Feuert die neuronale Zelle, so gibt sie einen Impuls über das Axon und die Synapsen an andere Neuronen weiter. Dort erhöht der Impuls das elektrische Potential, welches ab einer gewissen Höhe auch zu einem Feuern führt. Die Synapsen schwächen den ursprünglichen Impuls jedoch ab, so dass in der Regel mehrere eingehende Impulse benötigt werden, um die Zelle feuern zu lassen. Nach dem Abgeben eines Impulses benötigt das Neuron eine gewisse Pause, bis es wieder in der Lage ist, einen Impuls abzufeuern.

Aus dieser einfachen Struktur bildet die Natur komplexeste Netzwerke, in denen eine einzelne neuronale Zelle oft mit mehreren tausend anderen Neuronen verbunden ist [23].

Das menschliche Gehirn besteht aus ungefähr 10^{11} Neuronen und 10^{14} Synapsen. Verglichen mit der Geschwindigkeit von Computerbauteilen arbeiten die einzelnen neuronalen Zellen jedoch wesentlich langsamer. Ein Neuron benötigt zum Schalten mehrere Millisekunden, wohingegen Computerelemente millionenfach schneller arbeiten. Die ungeheure Leistungsfähigkeit des Gehirns im Vergleich zum Computer ist daher wahrscheinlich in dem wesentlich komplexeren Verbindungsnetzwerk zu suchen. Computerelemente sind nur mit sehr wenigen anderen Elementen verbunden, die Neuronen empfangen jedoch Informationen von mehreren tausend Neuronen und geben sie dann auch wieder an mehrere tausend Neuronen weiter. Das entstehende Netzwerk ist wesentlich komplexer und daher auch leistungsfähiger.

Modelliert werden diese natürlichen neuronalen Netzwerke unter anderem mit Hilfe von

Computerprogrammen. Die dabei entstehenden Programme werden auch “künstliche neuronale Netzwerke” (Artificial Neural Networks, kurz ANN) genannt [21, 23].

Der zentrale Gedanke ist das Modellieren des Feuerns mit Hilfe einer Schwellwertfunktion. McCulloch und Pitts führten hierzu folgende einfache Funktion zum Aktualisieren des Neurons n_i für den nächsten Zeitschritt $t + 1$ ein:

$$n_i(t + 1) = \Theta\left(\sum_j w_{ij}n_j(t) - \mu_i\right)$$

n_i kann hier entweder 0 oder 1 sein, womit der Zustand des Neurons (Feuern oder nicht Feuern) nachgebildet wird. Dazu wird $\Theta(x)$ wie folgt definiert (Heaviside Funktion):

$$\Theta(x) = \begin{cases} 1 & : x \geq 0 \\ 0 & : x < 0 \end{cases}$$

Die komplexen Abläufe in den Synapsen werden durch den Wert w_{ij} vereinfacht dargestellt, der nichts anderes als ein Gewicht für eine Verbindung zwischen zwei Neuronen darstellt. Alle eintreffenden Impulse n_j werden mit den zugehörigen Gewichten w_{ij} multipliziert und dann für das Neuron i aufsummiert. Erreicht die Summe den Schwellwert μ_i dann feuert das Neuron und gibt eine 1 zurück.

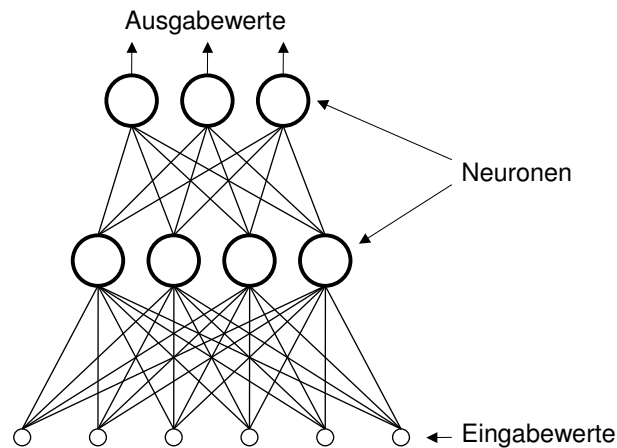


Abbildung 4.8: Künstliches neuronales Netzwerk

Obwohl das McCulloch-Pitts-Modell sehr einfach ist, ist es schon in der Lage, prinzipiell alle Berechnungen durchzuführen, die auch ein Computer auf herkömmliche Weise berechnen könnte. Natürlich muss ein eventuell sehr komplexes Netz aus Neuronen aufgebaut werden, und die Gewichte w_{ij} müssen passend gewählt werden. Ebenfalls muss angemerkt werden, dass nichts über die Effizienz der Berechnungen ausgesagt werden kann.

Verglichen mit den natürlichen neuronalen Netzwerken fallen bei diesem einfachen Modell viele Unterschiede auf. Die Forschung hat mittlerweile gezeigt, dass ein natürliches

Neuron nicht mit Hilfe von solch einfachen Modellen beschreibbar ist. Zum Beispiel entspricht das Modell des “Feuerns” nicht der Realität. Ein Neuron kann auch kontinuierlich auf die eingehenden Informationen reagieren. Ebenso entspricht das Ansteigen des elektrischen Potentials nicht einer einfachen Summenbildung. Verschiedene Arbeiten versuchen künstliche neuronale Netzwerke zu schaffen, die möglichst genau diese Eigenschaften der natürlichen Netzwerke nachbilden. Auf der anderen Seite werden für die Lösung von mathematischen Problemstellungen diese Eigenschaften eventuell gar nicht benötigt. Auf diese Aspekte wird hier jedoch nicht weiter eingegangen und auf die umfangreiche Literatur über ANN [23, 21] verwiesen.

Zwei grundsätzliche Probleme existieren bei der konkreten Anwendung eines solchen ANN. Das erste Problem ist, dass die Struktur des Netzwerks (Beispiel in Abbildung 4.8), also die Anzahl der Neuronen und die Verbindungen zwischen denselben, für ein gegebenes Problem erst festgelegt werden muss. In einer ersten Hochphase der Forschung um ANN konnte für einfache Netzwerke (sogenannte Perceptrons) ein garantiert terminierender Lernalgorithmus vorgestellt werden. Es bestand Hoffnung, dass dieser Algorithmus auch auf kompliziertere Netzwerktypen (Multi-Layer Perceptrons) erweitert werden könnte. So hätten konkrete Berechnungsprobleme einfach von einem Netzwerk gelernt werden können. Diese Hoffnung hat sich jedoch bald zerschlagen, und die einfachen Perceptrons waren für viele Berechnungsprobleme nicht leistungsfähig genug.

Die ANN-Forschung stagnierte daraufhin für einige Jahre, bis alte Forschungsergebnisse von Hebb wieder neu untersucht wurden. Hebb vermutete, dass die Änderung der Gewichte der Synapsen im menschlichen Gehirn proportional zur Korrelation der Aktivitäten der Neuronen sind. Korreliert die Aktivität zwischen zwei neuronalen Zellen, so sind auch die Verbindungen zwischen beiden stark. Diese Annahme wurde in der Zwischenzeit durch Untersuchungen für wesentliche Teile des Gehirns bestätigt.

Auf der Grundlage der Annahme von Hebb wurden verschiedene Lern- und Trainingsmethoden für künstliche neuronale Netzwerke entwickelt. Eine bekannte Technik ist der sogenannte Backpropagation-Algorithmus, mit dessen Hilfe auch komplexe Netzwerke trainiert werden können. Auch die Topology Representing Networks sind ein solcher Algorithmus, der die Annahme von Hebb für das Training des Netzwerks einsetzt.

4.5.2 Voronoi-Zellen und Delaunay-Triangulierung

Es existiert eine Reihe von verwandten Problemen, die auf ähnliche geometrische Fragestellungen reduziert werden können. Die Frage nach den k nächsten Nachbarn eines Punktes ist ein solches Problem. Ebenso die Frage nach dem Baum, dessen Eckpunkte eine Gruppe von Punkten sind, und dessen Verbindungen für diese Gruppe von Punkten minimale Länge aufweisen. Ein weiteres Beispiel ist das Triangulierungsproblem, bei dem N Punkte so verbunden werden, dass sich die Verbindungslinien nicht schneiden und innerhalb der konvexen Hülle der Punkte durch die Linien nur Dreiecke entstehen.

Alle diese Probleme tauchen in einer Vielzahl von Fragestellungen auf, zum Beispiel bei Computernetzwerken und deren Planung, bei Bildbearbeitung oder auch Spracherkennung. Die sogenannten Voronoi-Diagramme und Delaunay-Triangulierungen (Abbildung 4.9) stellen in diesem Zusammenhang zumindest einen Startpunkt für eine Lösung dar.

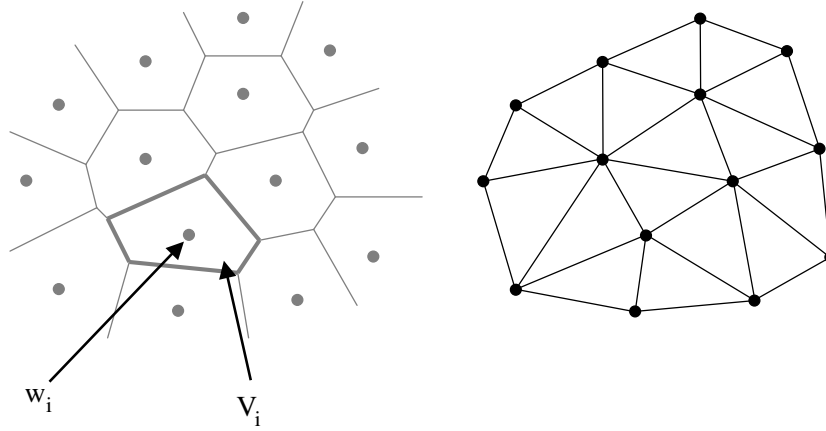


Abbildung 4.9: 2D-Voronoi-Diagramm und Delaunay-Triangulation

Definition 4.5.1 Voronoi-Diagramm

Ein Voronoi-Diagramm V_S einer Menge von Punkten $S = \{\mathbf{w}_1, \dots, \mathbf{w}_N\}$ mit $\mathbf{w}_i \in \mathbb{R}^D$ ist gegeben durch N D -dimensionale Polyeder, die sogenannten Voronoi Polyeder V_i , die wie folgt definiert werden:

V_i eines Punktes $\mathbf{w}_i \in S$ ist gegeben durch eine Menge von Punkten $\mathbf{v} \in \mathbb{R}^D$, deren Entfernung zu \mathbf{w}_i kleiner ist als zu allen anderen $\mathbf{w}_j \in S$, $j \neq i$, also

$$V_i = \{\mathbf{v} \in \mathbb{R}^D \mid \|\mathbf{v} - \mathbf{w}_i\| \leq \|\mathbf{v} - \mathbf{w}_j\| \text{ } j = 1, \dots, N\} \text{ } i = 1, \dots, N$$

◇

Die Voronoi-Polyeder stellen eine vollständige Zerlegung der Wertemenge \mathbb{R}^D dar, d.h. $\mathbb{R}^D = \bigcup_{i=1}^N V_i$.

Eine wichtige Eigenschaft der Voronoi Zellen ist das Vorhandensein von Vektoren $\mathbf{v} \in \mathbb{R}^D$, welche zu mehreren Zellen gehören. Diese Vektoren befinden sich genau auf den "Rändern" der Zellen. Alle diese Vektoren zusammen bilden einen Graphen. Im Fall $D = 2$ existieren Vektoren $\mathbf{v} \in \mathbb{R}^2$, die gleichzeitig zu drei Voronoi Zellen V_i gehören und die in dem Graphen die Eckpunkte bilden. Die Verbindungslinien entstehen durch die Vektoren \mathbf{v} , welche gleichzeitig zu genau zwei Zellen gehören.

Die Delaunay-Triangulierung entsteht, wenn alle $\mathbf{w}_i \in S$ verbunden werden, deren V_i eine gemeinsame Kante besitzen.

Definition 4.5.2 Delaunay-Triangulierung

Die Delaunay-Triangulierung D_s einer Menge von Punkten $S = \{\mathbf{w}_1, \dots, \mathbf{w}_N\}$ mit $\mathbf{w}_i \in \mathbb{R}^D$ ist definiert durch einen Graphen dessen Punkte die Punkte \mathbf{w}_i darstellen und dessen Verbindungsmatrix A mit $A_{ij} \in \{0, 1\}$, $i, j = 1, \dots, N$ wie folgt gegeben ist

$$A_{ij} = \begin{cases} 1 & V_i \cap V_j \neq \emptyset \\ 0 & \text{sonst} \end{cases}$$

◇

Zwei Punkte \mathbf{w}_i und \mathbf{w}_j des Graphen sind also verbunden, wenn die zugehörigen Voronoi-Zellen V_i und V_j benachbart sind.

4.5.3 Hebb-Regel erzeugt Delaunay-Triangulierung

Mit Hilfe der Voronoi-Zellen und der Delaunay-Triangulierung kann ein künstliches neuronales Netzwerk modelliert werden. Dieses neuronale Netzwerk ist etwas anders konstruiert als das oben beschriebene einfache McCulloch-Pitts Modell. Jede neuronale Zelle besitzt einen Gewichtsvektor \mathbf{w}_i und ist mit den anderen Zellen verbunden. Diese Verbindungen werden vergleichbar mit der Verbindungsmatrix A_{ij} der Delaunay-Triangulierung über eine Verbindungsmatrix C_{ij} mit $C_{ij} \in \mathbb{R}_0^+$ angegeben.

Im Bereich der neuronalen Netzwerke wird statt von der Voronoi-Zelle auch von dem “receptive field” eines Neurons gesprochen. Es werden Eingabevektoren an die neuronalen Zellen weitergegeben. Diese werden an jeder neuronalen Zelle i noch von einem Gewichtsvektor \mathbf{w}_i abgezogen, bevor sie in das Modell der Zelle eingerechnet werden. Eine Zelle reagiert typischerweise stärker, je geringer die Entfernung des Eingabevektors von dem Gewichtsvektor \mathbf{w}_i ist. Somit entsteht in der Eingabewertemenge eine Voronoi-Zelle, in der der Gewichtsvektor \mathbf{w}_i des Neurons i die kleinste Entfernung zu den dort vorhandenen Vektoren besitzt (verglichen mit den Gewichtsvektoren der anderen Neuronen). Wird ein Eingabevektor aus der Voronoi-Zelle V_i in das Netzwerk gegeben, so reagiert die Zelle i am stärksten.

Wie oben schon beschrieben, ist die Ermittlung geeigneter Parameter ein Problem beim Einsatz eines neuronalen Netzwerks. Martinetz und Schulten schlagen einen eigenen Trainingsalgorithmus vor, der die Vektoren \mathbf{w}_i sowie die Verbindungen C_{ij} iterativ bestimmt.

Der Algorithmus stützt sich auf die Annahme von Hebb, dass eine Änderung der Verbindung C_{ij} proportional zu der Aktivität von Zelle i und Zelle j ist, also:

$$\Delta C_{ij} \propto y_i y_j \quad (4.1)$$

wobei y_i die Ausgabe des Neurons i angibt. Wie oben schon beschrieben wird im hier vorliegenden ANN jedem Neuron ein Gewichtsvektor \mathbf{w}_i zugeordnet, der von einem Eingabevektor \mathbf{v} erst abgezogen wird. Auf das Ergebnis wird eine positive, stetige, monoton

fallende Funktion R angewendet, welche die Ausgabe des Neurons definiert.

$$y_i = R(\|\mathbf{v} - \mathbf{w}_i\|)$$

Also folgt für ΔC_{ij} :

$$\Delta C_{ij} \propto R(\|\mathbf{v} - \mathbf{w}_i\|)R(\|\mathbf{v} - \mathbf{w}_j\|)$$

Würde die Hebb-Regel in dieser einfachen Form in einem Trainingsalgorithmus eingesetzt, so würde einfach jedes Neuron mit jedem anderen Neuron verbunden. Die Stärke der Verbindung würde sich nach der Entfernung der zugehörigen Vektoren \mathbf{w}_i und \mathbf{w}_j richten.

Wird die Hebb-Regel jedoch so eingesetzt, dass die einzelnen Neuronen in Konkurrenz zueinander stehen und nur die beiden Zellen mit der höchsten Ausgabeaktivität eine Verbindungs-Update erfahren, so entsteht ein anderes Verbindungsnetzwerk.

$$\Delta C_{ij} \propto \begin{cases} y_i y_j & \text{mit } y_i y_j \geq y_k y_l \forall k, l = 1, \dots, N \\ 0 & \text{sonst} \end{cases}$$

Für $P(\mathbf{v}) \neq 0 \forall \mathbf{v} \in \mathbb{R}^D$ zeigen Martinetz und Schulten, dass gilt:

$$\Theta[C_{ij}(t \rightarrow \infty)] = A_{ij} \quad i, j = 1, \dots, N$$

wobei Θ die Heaviside-Funktion angibt. D.h. für eine ausreichend lange Trainingszeit, wird das Verbindungsnetzwerk äquivalent zur Delaunay-Triangulierung.

Zu beachten ist, dass die Wahrscheinlichkeit für alle $\mathbf{v} \in \mathbb{R}^D$ ungleich Null sein muss. Ein Trainingsalgorithmus, der obige Vorüberlegungen ausnutzt, wird das Netzwerk mit Hilfe von Vektoren $\mathbf{v} \in \mathbb{R}^D$ trainieren. Sinnvollerweise wird hierbei der Vektor \mathbf{v} mit der Wahrscheinlichkeit $P(\mathbf{v})$ ausgewählt. Ist $P(\mathbf{v})$ nur für eine Teilmenge $\mathbf{v} \in M$ ungleich Null, so wird das resultierende Netzwerk nur einen Teilgraph der Delaunay-Triangulierung darstellen.

4.5.4 Neural Gas Algorithm

Neben den Verbindungen C_{ij} ist ebenfalls die Wahl der Gewichtsvektoren \mathbf{w}_i von entscheidender Bedeutung. Sie werden genau wie die Verbindungen iterativ angepasst. Bei der Wahl der Update-Regel stützen sich Martinetz und Schulten wieder auf das Hebb-Postulat:

$$\Delta \mathbf{w}_i = \epsilon(\mathbf{v} - \mathbf{w}_i)f(k_i)$$

Hierbei gibt $(\mathbf{v} - \mathbf{w}_i)$ die präsynaptische und $f(k_i)$ die postsynaptische Erregung wieder. Hebb hatte ja genau diesen Zusammenhang postuliert, dass die Korrelation von prä- und

postsynaptischer Aktivität der Neuronen proportional die Änderung der Stärke der Verbindungen bedingt.

k_i gibt hier an, wieviele neuronale Zellen stärker als die untersuchte Zelle i auf den Eingangsvektor reagiert haben. Dies bedeutet, k_i Vektoren \mathbf{w}_j haben einen geringeren Abstand zu \mathbf{v} als \mathbf{w}_i . Um die Information k_i zu erhalten, müssen nach Eingabe des Vektors \mathbf{v} die Neuronen nach dem Abstand der zugehörigen \mathbf{w}_i sortiert werden.

Für die Funktion f gilt dabei $f(k_i) \in [0, 1]$. Sie wird wie folgt angegeben:

$$f(k_i) = e^{-\frac{k_i}{\lambda}}$$

Der Parameter λ gibt dabei ein Maß für die Anzahl der zu aktualisierenden Vektoren an. So werden nur die nah an \mathbf{v} liegenden \mathbf{w}_i aktualisiert, für die k_i klein ist. Der Parameter $\epsilon \in [0, 1]$ gibt ein generelles Maß für die Geschwindigkeit des Updates an, also, ob die \mathbf{w}_i bei jedem Schritt nur gering oder stark verändert werden sollen.

Insgesamt ergibt sich somit folgende Update-Regel:

$$\mathbf{w}_i^{neu} = \mathbf{w}_i^{alt} + \epsilon e^{-\frac{k_i}{\lambda}} (\mathbf{v} - \mathbf{w}_i^{alt})$$

Das iterative Anpassen der Gewichtsvektoren \mathbf{w}_i und das Aufstellen des Verbindungsnetzwerks C_{ij} kann gleichzeitig erfolgen. Problematisch ist jedoch das Verschieben der Vektoren \mathbf{w}_i für das Verbindungsnetzwerk. Vektoren \mathbf{w}_i , die zu einem frühen Zeitpunkt benachbart sind, können zu einem späteren Zeitpunkt diese Nachbarschaftsbeziehung nicht mehr besitzen. Daher wird für jede Verbindung C_{ij} ein Alter t_{ij} eingeführt. Bei jedem Update-Schritt wird das Alter aller Verbindungen erhöht und Verbindungen mit einem Alter $t_{ij} > T$ werden wieder verworfen (also $C_{ij} = 0$ gesetzt). T ist das maximale Alter einer Verbindung. Wird eine Verbindung während des Trainingsprozesses bestätigt, so wird das Alter t_{ij} wieder auf Null gesetzt. Auf diese Weise bleiben nur Verbindungen erhalten, deren zugehörige Vektoren \mathbf{w}_i am Ende der Trainingsphase des Netzwerks benachbart sind.

4.5.5 Delaunay-Triangulierung und Topology Preserving Maps

Betrachtet man die Verbindungen C_{ij} und ist $P(\mathbf{v})$ nur für eine Teilmenge M ungleich Null, so kann eine Abbildung Φ aufgestellt werden, welche jedem Wert aus M einen Index $i \in G$ (G sei die Indexmenge) zuordnet, der einen Vektor \mathbf{w}_i angibt:

$$\Phi : M \mapsto G, \mathbf{v} \in M \mapsto i^*(\mathbf{v}) \in G$$

wobei $i^*(\mathbf{v})$ wieder durch den minimalen Abstand des zugehörigen Vektors $\mathbf{w}_{i^*(\mathbf{v})}$ bestimmt wird:

$$\|\mathbf{w}_{i^*(\mathbf{v})} - \mathbf{v}\| \leq \|\mathbf{w}_i - \mathbf{v}\| \quad \forall i \in G$$

Ein Problem aus der ANN-Forschung stellen die sogenannten Topology Preserving Feature Maps dar. Eine solche Map entspricht obiger Abbildung Φ sowie deren Umkehrung Φ^{-1} . Zusätzlich muss jedoch noch die Bedingung gelten, dass diese Abbildungen nachbarschaftstreu sind. Dies bedeutet, dass benachbarte \mathbf{v} auch auf benachbarte $\mathbf{w}_{i^*(\mathbf{v})}$ abgebildet werden müssen. Gleiches muss auch für die Umkehrabbildung gelten. Genau dies ist jedoch in der Regel nicht leicht zu erreichen.

Eine solche Topology Preserving Feature Map ist in der Praxis sehr gut einsetzbar, da eine Analyse der Eingabevektoren \mathbf{v} durch die Analyse der aktivierten \mathbf{w}_i erreicht werden kann. Da die topologischen Beziehungen erhalten bleiben, sind topologische Eigenschaften der \mathbf{w}_i auch Eigenschaften der Vektoren \mathbf{v} . Die Topology Preserving Features Maps werden daher erfolgreich in der Spracherkennung oder Bildanalyse eingesetzt. Darüberhinaus konnte auch gezeigt werden, dass Teile des menschlichen Gehirns ebenfalls mit solchen Strukturen arbeiten.

Martinetz und Schulten zeigen in [36], dass der Teilgraph der Delaunay-Triangulierung, der bei der Anwendung des TRN-Algorithmus auf eine Teilmenge M entsteht, genau eine solche Topology Preserving Feature Map darstellt. Sie definieren dazu sogenannte maskierte Voronoi-Zellen, welche das Gegenstück zu diesem Teilgraph der Delaunay-Triangulierung darstellen. Weitere Details, eine formale Beschreibung sowie die zugehörigen Beweise können [36] entnommen werden.

4.5.6 Algorithmus

Diese Vorüberlegungen können durch einen äusserst kompakten Algorithmus umgesetzt werden.

1. Initialisiere die Vektoren $\mathbf{w}_i \in \mathbb{R}^D$, $i = 1, \dots, N$ (zufällig verteilt). Alle Verbindungen C_{ij} werden gleich Null, also “nicht verbunden”, gesetzt.
2. Wähle zufällig ein $\mathbf{v} \in \mathbb{R}^D$ (mit gleicher Wahrscheinlichkeit für alle \mathbf{v}).
3. Für jedes Neuron i müssen die k_i Neuronen bestimmt werden, für die gilt

$$\|\mathbf{v} - \mathbf{w}_j\| < \|\mathbf{v} - \mathbf{w}_i\|$$

4. Führe einen Update-Schritt für alle Vektoren \mathbf{w}_i durch:

$$\mathbf{w}_i^{neu} = \mathbf{w}_i^{alt} + \epsilon e^{-\frac{k_i}{\lambda}} (\mathbf{v} - \mathbf{w}_i^{alt})$$

5. Wenn $C_{i_0 i_1} = 0$, dann setze $C_{i_0 i_1}$ auf einen Wert > 0 , sowie $t_{i_0 i_1} = 0$. Hierbei ist i_0 der Index des Neurons mit $k_{i_0} = 0$. Analog gilt für Neuron i_1 : $k_{i_1} = 1$. Dieser Schritt verbindet Neuron i_0 mit Neuron i_1 . Waren sie schon verbunden, war also $C_{i_0 i_1} > 0$, dann setze $t_{i_0 i_1} = 0$.

6. Erhöhe das Alter aller Verbindungen der Zelle i_0 :

$$t_{i_0j} = t_{i_0j} + 1 \quad \forall j : C_{i_0j} > 0$$

7. Lösche alle Verbindungen von Zelle i_0 , welche das Zeitlimit überschritten haben. Setze also $C_{i_0j} = 0$ für alle j mit $C_{i_0j} > 0$ und $t_{i_0j} > T$. Fahre fort mit Schritt 2.

Zeitaufwendig ist Schritt 3. Dort werden die Entfernungen zwischen allen Vektoren \mathbf{w}_i und dem Trainingsvektor \mathbf{v} sortiert. Dies ist bei großen N problematisch. Jedoch wird $\Delta \mathbf{w}_i$ sehr klein, wenn $k_i \gg \lambda$. Wird diese kleine Änderung vernachlässigt, dann kann die Sortierung (bei kleinen λ) bei einem k abgebrochen werden, welches deutlich kleiner ist als N .

Martinetz und Schulten schlagen zudem vor, die Parameter λ , ϵ sowie T abhängig von der Laufzeit t des Algorithmus anzupassen. Naheliegender ist, zum Beispiel die Aktualisierungsgeschwindigkeit ϵ zu Beginn des Algorithmus recht groß zu wählen, da hier noch die zufällig gewählten initialen Positionen vorherrschen und großer Korrekturbedarf besteht. Später hat der Algorithmus jedoch schon recht gute Positionen gefunden, und die Anpassungen sollten eher kleiner sein. Daher werden die Parameter durch folgende Funktion über die Zeit t verändert:

$$g(t) = g_i(g_f/g_i)^{t/t_{max}}$$

wobei g durch λ , ϵ sowie T zu ersetzen ist. Die Parameter werden so in zwei Teilparameter g_i und g_f aufgeteilt, aus denen dann erst mit Hilfe von t und t_{max} (maximale Laufzeit) die endgültigen Parameter für den aktuellen Schritt des Algorithmus berechnet werden. Es wurden die Parameter $t_{max} = 1.000.000$; $\epsilon_i = 0,03$; $\epsilon_f = 0,001$; $\lambda_i = 0,2$; $\lambda_f = 0,02$ empirisch [69] bestimmt.

4.6 Erweitertes Verfahren

Der TRN-Algorithmus wird im Rahmen eines verbesserten Lösungsvorschlages verwendet, um die Kraftberechnungen deutlich zu beschleunigen und für das haptische Rendern einsetzbar zu machen.

4.6.1 Vektorquantisierung

Es wird im folgenden der TRN-Algorithmus zur Vektorquantisierung einer Dichteverteilung eingesetzt. Zur Demonstration wurde ein Molekül (PDB-Datei 2REC, siehe Abbildung 4.10-a) mit 2277 Atomen untersucht. Zunächst wurde die Struktur in eine Dichteverteilung

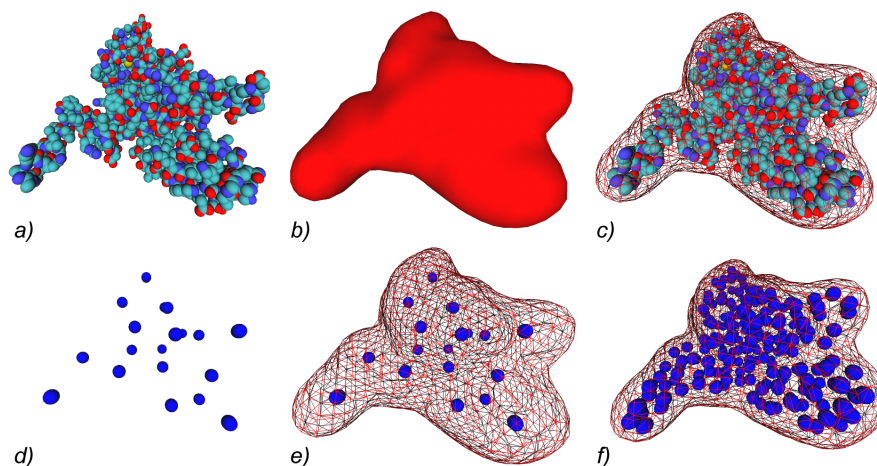


Abbildung 4.10: Vektorquantisierung (2REC Monomer)

mit 15 Ångström Auflösung umgewandelt. Der resultierende Volumendatensatz ist in Abbildung 4.10-b zu sehen (beziehungsweise in Abbildung 4.10-c als kombinierte Drahtgitterdarstellung mit dem ursprünglichen Molekül).

Anschließend wurde auf die Dichteverteilung der TRN-Algorithmus angewendet, wobei die Positionen von 20 Codebookvektoren berechnet wurden. Diese Codebookvektoren sind in Abbildung 4.10-d beziehungsweise in Abbildung 4.10-e (hier wieder als kombinierte Darstellung mit dem Volumen) zu sehen. Wie an der Abbildung 4.10-e festgestellt werden kann, hat der TRN-Algorithmus Codebookvektoren ermittelt, die die Eigenschaften des Volumendatensatzes gut repräsentieren. Dies läßt sich ebenfalls bei einem Vergleich der ursprünglichen Struktur mit den Codebookvektoren feststellen (siehe Abbildung 4.11).

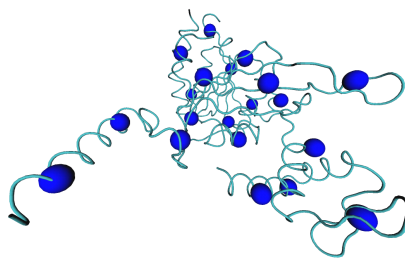


Abbildung 4.11: Codebookvektoren und 2REC Struktur (Tube-Modus)

Werden wesentlich mehr Codebookvektoren berechnet, so wird das Volumen immer besser repräsentiert. In Abbildung 4.10-f sind 200 Codebookvektoren dargestellt. Bei einem Vergleich mit Abbildung 4.10-c fällt die große Ähnlichkeit mit der ursprünglichen Struktur auf, obwohl auch hier die Anzahl der Codebookvektoren noch um über 90% kleiner ist als die Anzahl der Atome der Struktur.

4.6.2 Korrelationskoeffizient

Im Zentrum der Kraftberechnung steht die Berechnung der Korrelation, um die vom Benutzer gerade angesteuerte Abbildungsposition bewerten zu können. Wie schon im ersten Lösungsvorschlag beschrieben, wird hier die Korrelationsfunktion aus Abschnitt 4.3.3 verwendet. Die aktuelle Korrelation bei gegebenem R und \mathbf{T} ist dann:

$$C(R, \mathbf{T}) = \int \rho_{calc}(R, \mathbf{T}, \mathbf{r}) \rho_{em}(\mathbf{r}) d^3\mathbf{r}$$

Durch den Einsatz der Vektorquantisierung kann $\rho_{calc}(R, \mathbf{T}, \mathbf{r})$ durch den folgenden Term angenähert werden:

$$\rho_{calc}(R, \mathbf{T}, \mathbf{r}) \equiv \sum_{i=1}^k \delta(\mathbf{r} - \mathbf{w}_i(R, \mathbf{T}))$$

Hierbei ist $\delta(t)$ eine Dirac-Delta Funktion, für die gilt:

$$\delta(t) = \begin{cases} 0 & t \neq 0 \\ \infty & t = 0 \end{cases}$$

Der Term $\mathbf{w}_i(R, \mathbf{T})$ entspricht den Codebookvektoren, die durch den TRN-Algorithmus aus ρ_{calc} berechnet worden sind. Sie werden mit Hilfe der Parameter R und \mathbf{T} gedreht beziehungsweise verschoben. Dies kann wieder mit Hilfe von drei Rotationsmatrizen geschehen:

$$\mathbf{w}_i(R, \mathbf{T}) = M_\varphi M_\theta M_\phi(\mathbf{w}_i - \mathbf{O}) + \mathbf{O} + \mathbf{T}$$

wobei \mathbf{O} der Mittelpunkt der Codebookvektoren sei.

Setzt man den so aufgestellten ρ_{calc} -Term in die Koeffizienten-Gleichung ein, so ergibt sich:

$$C(R, \mathbf{T}) = \sum_{i=1}^k \int \delta(\mathbf{r} - \mathbf{w}_i(R, \mathbf{T})) \rho_{em}(\mathbf{r}) d\mathbf{r}$$

Da für die Dirac-Delta Funktionen gilt

$$\int \delta(t) dt = 1$$

$$\int f(t) \delta(t - t_0) dt = f(t_0)$$

kann die obige Gleichung weiter vereinfacht werden zu:

$$C(R, \mathbf{T}) = \sum_{i=1}^k \rho_{em}(\mathbf{w}_i(R, \mathbf{T}))$$

4.6.3 Beschreibung des Verfahrens

Mit Hilfe dieser vereinfachten Korrelationsfunktion kann ein neuer Lösungsvorschlag formuliert werden.

In der Vorbereitungsphase (Abbildung 4.12) wird der TRN Algorithmus auf die Molekülstrukturdaten angewendet. Die resultierenden Codebookvektoren werden genauso wie die niedrigaufgelöste Dichteverteilung in der Hauptschleife (Abbildung 4.13) benötigt.

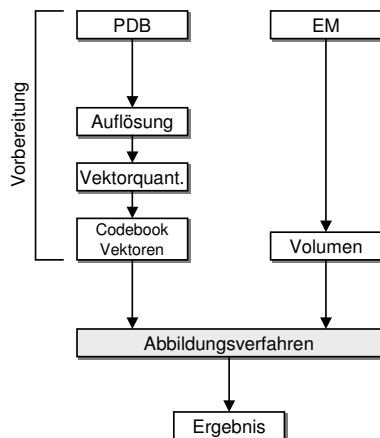


Abbildung 4.12: Vorbereitungsphase

Die Hauptschleife des Verfahrens ist durch die Nutzung der Codebookvektoren wesentlich kompakter geworden. Die aufwendigen Manipulationen und Untersuchungen der Volumendatensätze sind entfallen. Es wird nur noch der aktuelle Status des Phantom abgefragt und somit die Parameter R und T für die obige Korrelationsfunktion ermittelt.

Diese neue Berechnungsvorschrift ist im Gegensatz zum ersten Lösungsvorschlag auch mit der für das haptische Rendern notwendigen Geschwindigkeit durchführbar. In dem Kapitel 6 wird die Implementierung des Verfahrens beschrieben und dann auch auf die Ergebnisse eingegangen. Dort wird diese Behauptung bezüglich der Geschwindigkeit durch konkrete Beispielrechnungen und die dabei erreichten Kraft-Update-Raten belegt.

In der bisherigen Beschreibung der Methode wurde die eigentliche Kraftberechnung noch ausgespart. Sie wird nun im folgenden Abschnitt besprochen sowie ihre Anwendung im Falle des haptischen Gerätes der Firma SensAble.

4.7 Kraftberechnung

Die Grundlage für die Kraftberechnung bildet die physikalische Modellierung eines starren Körpers in einem konservativen Kraftfeld. Ein solcher Körper kann mit Rahmen der klassischen Mechanik beschrieben werden. Isaak Newton hat 1689 in seinem Buch “Philosophia Naturalis Principia Mathematica” die erste geschlossene Theorie der Mechanik

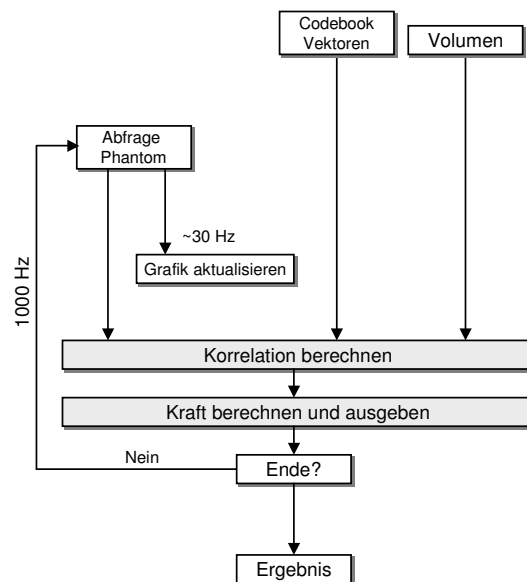


Abbildung 4.13: Hauptschleife

aufgestellt. Im folgenden werden die für das Verständnis des Verhaltens eines starren Körpers notwendigen Teilbausteine dieser Theorie eingeführt.

4.7.1 Mechanik eines Teilchens

Ein starrer Körper besteht aus einer Menge von Masseteilchen. Die Mechanik eines einzelnen Teilchens läßt sich aus den Newtonschen Gesetzen herleiten:

Lex Prima “Jeder Körper verharrt in seinem Zustand der Ruhe oder der gleichförmig geradlinigen Bewegung, wenn er nicht durch einwirkende Kräfte gezwungen wird, seinen Bewegungszustand zu ändern.”

Der Körper ist hier das einzelne Masseteilchen. Der Bewegungszustand wird auch mit dem Begriff Impuls beschrieben. Das Gesetz besagt somit, dass der Impuls konstant ist, wenn keine Kraft auf das Masseteilchen einwirkt. Es ist dann

$$\mathbf{p} = m\mathbf{v}$$

konstant, wobei \mathbf{p} der lineare Impuls, m die Masse und \mathbf{v} die Geschwindigkeit ist.

Lex Secunda “Die Änderung der Bewegungsgröße ist der Einwirkung der bewegenden Kraft proportional und geschieht nach der Richtung derjenigen geraden Linie, nach welcher jene Kraft wirkt.”

Die Änderung der Bewegungsgröße ist mathematisch die Ableitung von \mathbf{p} über der Zeit t . Also besagt das Gesetz:

$$\frac{d}{dt}\mathbf{p} = \mathbf{F}$$

Ein Masseteilchen ändert normalerweise seine Masse nicht über der Zeit, so daß folgt:

$$\frac{d}{dt}\mathbf{p} = m \frac{d}{dt}\mathbf{v} = m\dot{\mathbf{v}} = \mathbf{F}$$

Hierbei ist $\dot{\mathbf{v}}$ die Änderung der Geschwindigkeit \mathbf{v} über der Zeit, die Beschleunigung.

Neben dem linearen Impuls \mathbf{p} kann auch der Drehimpuls \mathbf{l} definiert werden. Das Teilchen wird um einen Punkt \mathbf{O} gedreht, wobei \mathbf{r} der Vektor von \mathbf{O} zum Teilchen sei:

$$\mathbf{l} = \mathbf{r} \times \mathbf{p}$$

Wird die Änderung des Drehimpulses über der Zeit untersucht, so ergibt sich

$$\frac{d}{dt}\mathbf{l} = \frac{d}{dt}(\mathbf{r} \times \mathbf{p}) = \left[\left(\frac{d}{dt}\mathbf{r}\right) \times \mathbf{p}\right] + \left[\mathbf{r} \times \left(\frac{d}{dt}\mathbf{p}\right)\right] = \mathbf{v} \times \mathbf{p} + \mathbf{r} \times \left(\frac{d}{dt}\mathbf{p}\right)$$

wobei $\mathbf{p} = m\mathbf{v}$, also gilt

$$\mathbf{v} \times m\mathbf{v} + \mathbf{r} \times \left(\frac{d}{dt}m\mathbf{v}\right) = \mathbf{r} \times \left(\frac{d}{dt}m\mathbf{v}\right) = \mathbf{r} \times m\dot{\mathbf{v}} = \mathbf{r} \times \mathbf{F} = \mathbf{N}$$

Es entsteht die Größe \mathbf{N} , das sogenannte Drehmoment. \mathbf{N} ist also analog zur Kraft \mathbf{F} definiert, welche der zeitlichen Änderung des linearen Impulses entspricht. Das Drehmoment beantwortet die Frage, was passiert, wenn der Abstand des Masseteilchens zum Bezugspunkt \mathbf{O} konstant bleiben muss und trotzdem eine Kraft \mathbf{F} auf das Teilchen einwirkt. Es entsteht eine Kraft, welche das Teilchen um einen Drehvektor bewegen möchte. Das Drehmoment entspricht genau diesem Vektor (Abbildung 4.14). Es ist relativ zu \mathbf{O} definiert und seine Länge stellt die Höhe der "Drehkraft" dar.

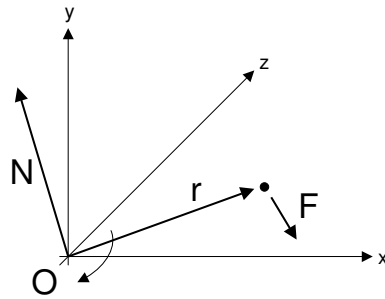


Abbildung 4.14: Drehmoment

Geometrisch bedeutet das Vektorkreuzprodukt, dass der Resultatvektor senkrecht auf der Ebene steht, die von den Argumentvektoren aufgespannt wird. Für $\mathbf{N} = \mathbf{r} \times \mathbf{F}$ heisst das, dass der Drehmoment-Vektor senkrecht auf der von \mathbf{r} und \mathbf{F} gebildeten Ebene steht.

Der Massenpunkt erfährt bei einer Drehung um den Punkt \mathbf{O} Positionsänderungen, die auf einer festen Bahn liegen. Daher ist oft mehr die Änderung des Winkels ϕ als die Änderung der Position \mathbf{r} von Interesse.

Für die Position \mathbf{r} wurde eine Größe \mathbf{v} eingeführt, die die zeitliche Änderung von \mathbf{r} erfasste. Eine analoge Größe kann auch für den Winkel ϕ definiert werden

$$\omega = \frac{d}{dt}\phi$$

die sogenannte Winkelgeschwindigkeit. Auch sie kann über die Zeit eine Änderung erfahren

$$\alpha = \frac{d}{dt}\omega$$

was die Winkelbeschleunigung α bezeichnet.

Potential und konservative Kraftfelder

Wird ein Massenpunkt durch eine Kraft bewegt, so wird eine Arbeit verrichtet

$$W = \mathbf{F} \cdot \mathbf{s}$$

Die Arbeit ist also gleich Kraft mal dem Weg s . Wird ein Masseteilchen durch ein Kraftfeld $\mathbf{F}(\mathbf{r})$ von einem Punkt \mathbf{r}_1 zu einem Punkt \mathbf{r}_2 bewegt, so wird folgende Arbeit verrichtet:

$$W(\mathbf{r}_1, \mathbf{r}_2) = \int_{\mathbf{r}_1}^{\mathbf{r}_2} \mathbf{F}(\mathbf{r}) \, d\mathbf{r}$$

Durchläuft das Masseteilchen das Kraftfeld auf einer geschlossenen Kurve C , so gilt

$$W_C = \oint_C \mathbf{F}(\mathbf{r}) \cdot d\mathbf{r}$$

Ein Kraftfeld $\mathbf{F}(\mathbf{r})$ wird konservativ genannt, wenn für beliebige geschlossene Kurven C die verrichtete Arbeit W_C immer Null ist. Dies bedeutet, dass für jedes Teilstück der Kurve C , für das die Arbeit positiv ist, auch wieder ein Teilstück existieren muss, auf dem die Arbeit negativ ist. Treten Reibungskräfte auf, dann ist die Arbeit für diese Kräfte jedoch immer positiv und das Integral über die Kurve kann nicht zu Null werden. Ein Kraftfeld, in dem solche Kräfte auftreten, wäre somit nicht konservativ.

Ist das Kraftfeld konservativ, so ist $W(\mathbf{r}_1, \mathbf{r}_2)$ unabhängig vom Weg zwischen \mathbf{r}_1 und \mathbf{r}_2 . Also muss es möglich sein, W auch nur als Änderung einer Größe zu beschreiben, die von der Lage von \mathbf{r}_1 beziehungsweise \mathbf{r}_2 abhängt:

$$W(\mathbf{r}_1, \mathbf{r}_2) = V(\mathbf{r}_1) - V(\mathbf{r}_2)$$

Diese Größe wird Potential oder potentielle Energie genannt. Für ein konservatives Kraftfeld F gilt:

$$\mathbf{F}(\mathbf{r}) = -\nabla V(\mathbf{r})$$

wobei ∇ der Nabla-Operator ist:

$$\nabla f = \begin{pmatrix} \frac{d}{dx} f \\ \frac{d}{dy} f \\ \frac{d}{dz} f \end{pmatrix}$$

Wird die Arbeit W betrachtet, die durch ein Kraftfeld an einem Teilchen geleistet wird, wenn es sich von Punkt \mathbf{r}_1 nach Punkt \mathbf{r}_2 bewegt:

$$W(\mathbf{r}_1, \mathbf{r}_2) = \int_{\mathbf{r}_1}^{\mathbf{r}_2} \mathbf{F}(\mathbf{r}) d\mathbf{r}$$

Wie oben schon beschrieben wird im hier betrachteten Fall ein Masseteilchen eine über die Zeit konstante Masse besitzen. So kann $\mathbf{F} = m \frac{d}{dt} \mathbf{v}$ eingesetzt werden

$$W(\mathbf{r}_1, \mathbf{r}_2) = m \int_{\mathbf{r}_1}^{\mathbf{r}_2} \frac{d}{dt} \mathbf{v} d\mathbf{r}$$

Da $\mathbf{v} = \frac{d\mathbf{r}}{dt}$, gilt ausserdem

$$W(\mathbf{r}_1, \mathbf{r}_2) = \int_{\mathbf{r}_1}^{\mathbf{r}_2} m \frac{d\mathbf{v}}{dt} \mathbf{v} dt = \frac{m}{2} \int_{\mathbf{r}_1}^{\mathbf{r}_2} \frac{d\mathbf{v}^2}{dt} dt = \frac{m}{2} (\mathbf{v}_{\mathbf{r}_2}^2 - \mathbf{v}_{\mathbf{r}_1}^2)$$

Die daraus ableitbare einzelne Größe

$$E_{kin}(\mathbf{r}) = \frac{1}{2} m \mathbf{v}_{\mathbf{r}}^2$$

wird die kinetische Energie des Teilchens genannt. Für die Arbeit folgt daraus:

$$W(\mathbf{r}_1, \mathbf{r}_2) = E_{kin}(\mathbf{r}_2) - E_{kin}(\mathbf{r}_1)$$

Da für die potentielle Energie gilt:

$$W(\mathbf{r}_1, \mathbf{r}_2) = V(\mathbf{r}_1) - V(\mathbf{r}_2)$$

gilt auch

$$E_{kin}(\mathbf{r}_2) - E_{kin}(\mathbf{r}_1) = V(\mathbf{r}_1) - V(\mathbf{r}_2)$$

Die gesamte Energie eines Teilchens berechnet sich aus der Summe von kinetischer und potentieller Energie

$$E(\mathbf{r}) = E_{kin}(\mathbf{r}) + V(\mathbf{r})$$

Die Energie eines Teilchens bleibt in einem konservativen Kraftfeld immer erhalten. Es wird bei einer Bewegung nur potentielle Energie in kinetische Energie umgewandelt (und umgekehrt), die Summe aus beiden bleibt aber gleich.

Wegen

$$\mathbf{F}(\mathbf{r}) = -\nabla V(\mathbf{r})$$

gilt ausserdem, dass die Kraft an einem Punkt \mathbf{r} im Kraftfeld immer in Richtung des negativen Gradienten der potentiellen Energie zeigt. Bewegt sich das Teilchen also in Richtung dieser Kraft, so wird die potentielle Energie minimiert (sie wird in kinetische Energie umgesetzt).

4.7.2 Teilchensysteme

Mit den gerade eingeführten Teilchen können auch Teilchensysteme definiert werden. Hier steht eine Menge von N Masseteilchen in einer Beziehung zueinander. Die einzelnen Teilchen des Systems besitzen Positionen \mathbf{r}_i , die sich über die Zeit ändern. Es wirken äußere Kräfte eines Kraftfeldes $\mathbf{F}(\mathbf{r}_i)$ auf die Teilchen ein, die unabhängige Bewegungen bewirken. Da die einzelnen Teilchen aber zu einem Gesamtsystem gehören, stehen sie in Beziehung zueinander. Daher treten auch noch innere Kräfte $\mathbf{F}(i, j)$ zwischen zwei Teilchen i und j auf.

Die Newtonsche Bewegungsgleichung $\mathbf{F} = \frac{d}{dt}\mathbf{p} = \dot{\mathbf{p}}$ lautet für ein Teilchen i in einem System also:

$$\sum_j \mathbf{F}(j, i) + \mathbf{F}(\mathbf{r}_i) = \dot{\mathbf{p}}_i$$

Die inneren Kräfte $\mathbf{F}(i, j)$ sollen dem dritten Newtonschen Gesetz gehorchen:

Lex Tertia “Die Wirkung ist stets der Gegenwirkung gleich; oder, die Wirkungen zweier Körper aufeinander sind stets gleich und von entgegengesetzter Richtung.”

Also gilt $\mathbf{F}(i, j) = -\mathbf{F}(j, i)$. Wird über die einzelnen Bewegungsgleichungen summiert, so ergibt sich:

$$\sum_{i,j} \mathbf{F}(j, i) + \sum_i \mathbf{F}(\mathbf{r}_i) = \sum_i \dot{\mathbf{p}}_i = \sum_i \frac{d}{dt} m_i \mathbf{v}_i = \sum_i m_i \frac{d^2}{dt^2} \mathbf{r}_i = \frac{d^2}{dt^2} \sum_i m_i \mathbf{r}_i$$

Der erste Term der linken Seite der Gleichung $\sum_{i,j} \mathbf{F}(j, i)$ kann als Summe der Paare $\mathbf{F}(j, i) + \mathbf{F}(i, j)$ umformuliert werden. Aufgrund des dritten Newtonschen Gesetzes fällt dieser Term weg.

Die rechte Seite kann weiter vereinfacht werden, wenn der Schwerpunkt des Gesamtsystems eingeführt wird. Die Masse des Systems läßt sich aus der Summe der Massen der einzelnen Teilchen bestimmen

$$M = \sum_i m_i$$

woraus sich dann der Schwerpunkt des Systems wie folgt aus den Positionen und Massen der Teilchen berechnet:

$$\mathbf{R} = \frac{\sum_i m_i \mathbf{r}_i}{M}$$

Wird in die obige Gleichung

$$\sum_i \mathbf{F}(\mathbf{r}_i) = \frac{d^2}{dt^2} \sum_i m_i \mathbf{r}_i$$

der Schwerpunkt eingesetzt, so ergibt sich

$$\sum_i \mathbf{F}(\mathbf{r}_i) = M \frac{d^2}{dt^2} \mathbf{R} = M \ddot{\mathbf{R}}$$

Der Schwerpunkt des Systems bewegt sich also durch die Wirkung der äusseren Kräfte auf die Gesamtmasse. Innere Kräfte spielen bei der Bewegung des Schwerpunktes keine Rolle.

Aus den linearen Einzelimpulsen der Teilchen kann der lineare Gesamtimpuls des Systems abgeleitet werden:

$$\mathbf{P} = \sum_i \mathbf{p}_i = \sum_i m_i \mathbf{v}_i = \sum_i m_i \dot{\mathbf{r}}_i = M \dot{\mathbf{R}}$$

Auch der Gesamtdrehimpuls (relativ zu einem Bezugspunkt \mathbf{O} und nicht relativ zum Schwerpunkt \mathbf{R}) wird über die Teildrehimpulse wie folgt berechnet:

$$\mathbf{L} = \sum_i \mathbf{r}_i \times \mathbf{p}_i = \sum_i \mathbf{l}_i$$

Sei \mathbf{R} der Vektor von \mathbf{O} zum Schwerpunkt (Abbildung 4.15), dann können die Teilchenpositionen \mathbf{r}_i auch relativ zu \mathbf{R} definiert werden

$$\mathbf{r}_i = \mathbf{r}'_i + \mathbf{R}$$

analog gilt für die Teilchengeschwindigkeit

$$\mathbf{v}_i = \mathbf{v}'_i + \mathbf{v}$$

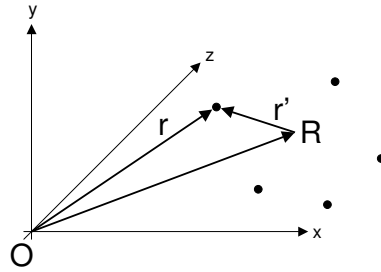


Abbildung 4.15: Schwerpunkt

mit $\mathbf{v} = \frac{d}{dt}\mathbf{R}$ (Geschwindigkeit des Schwerpunktes bezogen auf O). Der Drehimpuls \mathbf{L} kann umgeformt werden zu

$$\mathbf{L} = \mathbf{R} \times M\mathbf{v} + \sum_i \mathbf{r}'_i \times \mathbf{p}'_i$$

Er besteht also aus einem Teil, welcher relativ zu O definiert ist, da \mathbf{R} vom Bezugspunkt O abhängt. Der rechte Term ist relativ zum Schwerpunkt R definiert und hängt nicht von O ab.

Hieraus kann die Bewegungsgleichung

$$\mathbf{N} = \frac{d}{dt}\mathbf{L}$$

hergeleitet werden, wobei N das Gesamtdrehmoment des Systems ist. Dieses entspricht auch der Summe der Drehmomente der Teilchen des Systems, also

$$\mathbf{N} = \sum_i \mathbf{N}_i$$

Die kinetische Energie eines solchen Teilchensystems berechnet sich wie folgt

$$E_{kin} = \sum_i E_{kin}(i) = \frac{1}{2} \sum_i m_i \mathbf{v}_i^2$$

Wird hier wieder berücksichtigt, dass die \mathbf{v}_i relativ zum Bezugspunkt O definiert ist, dann kann die Gleichung wie folgt umgestellt werden:

$$E_{kin} = \frac{1}{2} M \mathbf{v}^2 + \frac{1}{2} \sum_i m_i \mathbf{v}'_i{}^2$$

Analog zum Gesamtdrehimpuls zerfällt die kinetische Energie in einen zu O und einen zu R relativen Teil.

Im Teilchensystem existieren innere und äussere Kräfte, so dass auch zwei verschiedene Potentiale auftreten. V_i beschreibt die potentielle Energie eines einzelnen Teilchens und V_{ij} die zwischen zwei Teilchen existierende potentielle Energie. Das Gesamtpotential berechnet sich daraus wie folgt:

$$V = \sum_i V_i + \frac{1}{2} \sum_{i,j} V_{ij}$$

Für die inneren und äusseren Kräfte gilt:

$$\mathbf{F}(i, j) = -\nabla V_{ij}$$

und

$$\mathbf{F}(r_i) = -\nabla V_i$$

Die Gesamtenergie des Systems ist $E = E_{kin} + V$ und für diese Energie gilt analog zum einzelnen Teilchen wieder die Erhaltungsbedingung.

4.7.3 Mechanik eines starren Körpers

Der starre Körper ist ein Sonderfall des gerade eingeführten Teilchensystems. Die Besonderheit des starren Körpers liegt darin, dass die Teilchen einen konstanten Abstand aufweisen, der sich über die Zeit nicht ändert. Es bewegt sich also nur der gesamte starre Körper, die einzelnen Teilchen verändern ihre Position nicht relativ zueinander. Da die Teilchen starr miteinander verbunden sind, sind die inneren Kräfte genauso groß, dass sie sich aufheben (anderenfalls würde sich der Abstand der Teilchen durch die inneren Kräfte ändern). Die inneren Kräfte werden daher nicht mehr betrachtet.

An der Bewegungsgleichung für den Schwerpunkt ändert sich nichts, da

$$\mathbf{F} = \sum_i \mathbf{F}(r_i) = M\ddot{\mathbf{R}}$$

auch für das normale Teilchensystem nicht von den inneren Kräften abhängt. Also bleibt auch der Gesamtimpuls \mathbf{P} gleich.

Durch das Verschwinden der relativen Bewegung der Teilchen vereinfachen sich aber einige andere Gleichungen. Der Drehimpuls \mathbf{L} wird zu

$$\mathbf{L} = \mathbf{R} \times M\mathbf{v}$$

da der Teil, der nur vom Schwerpunkt \mathbf{R} abhängt, wegfallen kann. Das Drehmoment $\mathbf{N} = \frac{d}{dt}\mathbf{L}$ ist dann:

$$\mathbf{N} = \frac{d}{dt}(\mathbf{R} \times M\mathbf{v}) = \dot{\mathbf{R}} \times M\mathbf{v} + \mathbf{R} \times M\dot{\mathbf{v}} = \mathbf{v} \times M\mathbf{v} + \mathbf{R} \times M\ddot{\mathbf{R}} = \mathbf{R} \times \mathbf{F}$$

Ebenso fällt in der Berechnung der kinetischen Energie der zu \mathbf{R} relative Teil weg:

$$E_{kin} = \frac{1}{2} M \mathbf{v}^2$$

Auch die Berechnung des Gesamtpotentials vereinfacht sich, da die potentielle Energie V_{ij} zwischen zwei Teilchen i und j nicht mehr beachtet werden muss:

$$V = \sum_i V_i$$

Ebenso gilt für einen starren Körper in einem konservativen Kraftfeld:

$$\mathbf{F} = -\nabla V$$

Die Kraft \mathbf{F} , die am Schwerpunkt \mathbf{R} eines starren Körpers in einem konservativen Kraftfeld wirkt, ist also immer entlang des negativen Gradienten der potentiellen Energie ausgerichtet. Wirkt keine andere Kraft, so wird der Körper sich also in Richtung der Kraft \mathbf{F} bewegen und sich so in ein Minimum der potentiellen Energie bewegen. Die potentielle Energie wird bei diesem Prozeß in kinetische Energie umgesetzt.

4.7.4 Anwendung der Gesetze der Mechanik zur Kraftberechnung

Die formulierten elementaren Gesetze der Mechanik bilden nun die Basis für die Kraftberechnung des haptischen Renderns. Wie schon in Abschnitt 4.4 beschrieben, soll der Benutzer im Rahmen des interaktiven Einpassprozesses in Richtung einer besseren Abbildungsposition gezogen werden. Hierbei soll “besser” gleichbedeutend mit einer Position und Orientierung sein, für die die Bewertungsfunktion C ansteigt.

Wie in Abschnitt 4.6 beschrieben, wird das Molekül durch eine Menge von Merkmalsvektoren ersetzt. Diese sogenannten Codebookvektoren können als Masseteilchen interpretiert werden. Da in der vorliegenden Arbeit ein “Rigid-Body Docking” durchgeführt werden soll, verändern diese ihre Positionen relativ zueinander nicht. Dies bedeutet, dass die gesamte Menge der Codebookvektoren einen starren Körper bildet.

Wenn der Benutzer wie in Abschnitt 4.6 beschrieben das Molekül interaktiv bewegt, dann setzt er dazu das haptische Gerät SensAble Phantom ein (siehe dazu auch Kapitel 3.3). Dieses Gerät besitzt einen Stift, welchen der Benutzer in der Hand führt. Die Position des Stiftes und seine Orientierung werden vom Gerät gemessen. Gleichzeitig kann das Gerät auf die Spitze des Stiftes eine Kraft ausüben. Das können translativ Kräfte als auch Drehmomente (nur Phantom 6DOF) sein. Der Benutzer bewegt das Handstück des Phantoms, und die jeweilige Orientierung und Position werden auf das abzubildende Molekül übertragen.

Bezogen auf den starren Körper (also das Molekül beziehungsweise die Codebookvektoren) bedeutet die Bewegung eine Translation oder Drehung bezüglich seines Schwerpunktes \mathbf{R} . Die Kräfte, die der Benutzer spürt, sollten daher den Kräften des starren Körpers bezüglich seines Massenschwerpunktes entsprechen.

Konservatives Kraftfeld

Der starre Körper befindet sich in einem konservativen Kraftfeld, wobei auf jeden Codebookvektor \mathbf{w}_i eine Kraft $\mathbf{F}(\mathbf{w}_i)$ einwirkt. Für die Berechnung dieser Kraft soll die Korrelationsfunktion

$$C(R, \mathbf{T}) = \sum_{i=1}^k \rho_{em}(\mathbf{w}_i(R, \mathbf{T}))$$

eingesetzt werden. Wird die dem Kraftfeld zugrundeliegende Potentialfunktion $V(\mathbf{w}_i)$ wie folgt gewählt

$$V(\mathbf{w}_i) = -\rho_{em}(\mathbf{w}_i)$$

so ergibt sich für das Potential des Gesamtkörpers

$$V = \sum_i V(\mathbf{w}_i) = \sum_i -\rho_{em}(\mathbf{w}_i) = -C$$

Es gilt für die Kraft, welche an einem Teilchen des starren Körpers angreift

$$\mathbf{F}(\mathbf{w}_i) = -\nabla V(\mathbf{w}_i) = \nabla \rho_{em}(\mathbf{w}_i)$$

Dies bedeutet, dass ein einzelnes Teilchen in Richtung des positiven Gradienten seiner Korrelation gezogen wird. Für den gesamten Körper bedeutet dies

$$\mathbf{F} = \sum_i \mathbf{F}(\mathbf{w}_i)$$

Diese Gesamtkraft wirkt auf das Massezentrum des Körpers als Translationskraft.

Drehmoment

Im Gegensatz zu einem einzelnen Teilchen wird der Körper von dem Kraftfeld auch durch ein Drehmoment beeinflusst. Wie in Kapitel 3.3 beschrieben, stellt das SensAble Phantom 6DOF auch die Möglichkeit der Ausgabe des Drehmoments zur Verfügung. Die Spitze des Handstücks bildet den Bezugspunkt O für das ausgegebene Drehmoment. Der Benutzer führt das abzubildende Molekül mit dem Handstück des Phantoms, wobei die Spitze des

Handstücks den Schwerpunkt der Codebookvektoren darstellt. Da diese Spitze auch den Bezugspunkt für das ausgegebene Drehmoment darstellt, ist also das Drehmoment relativ zum Schwerpunkt \mathbf{R} gesucht:

$$\mathbf{N} = \sum_i \mathbf{w}'_i \times \mathbf{F}(\mathbf{w}_i)$$

Die Koordinaten \mathbf{w}'_i sind relativ zum Schwerpunkt \mathbf{R} definiert, im Gegensatz zu den Koordinaten \mathbf{w}_i , die relativ zum Ursprung angegeben werden. Es können beide Koordinaten leicht ineinander überführt werden:

$$\mathbf{w}_i = \mathbf{w}'_i + \mathbf{R}$$

Obige Gleichung berechnet also das Drehmoment \mathbf{N} relativ zum Schwerpunkt \mathbf{R} des Moleküls. Dies ist das gesuchte Drehmoment, da die Spitze des Handstücks dem Schwerpunkt \mathbf{R} entspricht und das ausgegebene Drehmoment relativ zu eben dieser Spitze angegeben wird.

Bei der physikalischen Modellierung des haptischen Renderns ist zu beachten, dass es nicht darum geht, eine selbständig ablaufende Simulation zu entwickeln. Das Molekül bewegt sich im System nicht autonom, sondern die aktuelle Position wird von dem haptischen Gerät vorgeschrieben. Für diese Position und Orientierung wird mit Hilfe der mechanischen Gesetze eine Kraft und ein Drehmoment berechnet.

4.8 SensAble Phantom

In der vorliegenden Arbeit wird zum haptischen Rendern das SensAble Phantom eingesetzt. Eine genauere Beschreibung findet sich in Kapitel 3.3. Das Gerät hat zwei Aufgaben. Zunächst muss die aktuelle Position und Orientierung des Handstücks und damit des Moleküls (beziehungsweise der Codebookvektoren) gemessen werden. Darüberhinaus ist das Phantom aber nicht nur ein dreidimensionales Eingabegerät, sondern kann auch Kräfte als auch Drehmomente ausgeben.

4.8.1 Koordinaten

Zur Laufzeit des Programms müssen die Parameter \mathbf{w}_i , \mathbf{w}'_i und \mathbf{R} zur Verfügung stehen, damit die Kraft \mathbf{F} und das Drehmoment \mathbf{N} berechnet werden können. Zunächst werden in einem Vorbereitungsschritt die Codebookvektoren \mathbf{w}_i aus der Molekülstruktur mit Hilfe des TRN Algorithmus berechnet. Für diese Codebookvektoren kann dann der Schwerpunkt \mathbf{R} ermittelt werden.

Es bietet sich weiterhin an, im Vorbereitungsschritt die Codebookvektoren in den Ursprung zu verschieben. Durch eine Verschiebung aller \mathbf{w}_i um \mathbf{R} kann dies erreicht werden. Die daraus resultierenden Codebookvektoren stellen die zur Laufzeit notwendigen \mathbf{w}'_i dar.

Wenn der Benutzer das Handstück des Phantoms bewegt, so übermittelt das Gerät an das Programm eine (4×4) -Matrix. Diese Matrix stellt den Zustand des Handstücks dar und beschreibt sowohl Translation als auch Rotation. Eine (4×4) -Matrix stellt eine Kombination aus einer Rotations- beziehungsweise Skalierungsmatrix und einem Translationsvektor dar:

$$\begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Werden passende vierdimensionale Vektoren mit einer vierten Komponente verwendet, so kann mit einer einzelnen Matrix-Vektormultiplikation ein Vektor sowohl gedreht als auch verschoben werden. Eine solche Matrix kann so Orientierung und Position eines dreidimensionalen Objektes beschreiben und stellt den Übergang von dem lokalen Koordinatensystem des Objektes in das umgebende Weltkoordinatensystem dar.

Das SensAble Phantom gibt über seine Software eine solche (4×4) -Matrix $M_{Phantom}$ zur Beschreibung der Lage und Orientierung des Handstücks relativ zu seinem eigenen Weltkoordinatensystem aus.

Um dieses Weltkoordinatensystem aufstellen zu können, muss das Phantom zu Beginn einer Programmnutzung neu kalibriert werden. Dazu hält der Benutzer den Stift in einer Ausgangsposition fest, welche nach dem Bestätigen der Kalibrierungsmeldung als neuer Ursprung des Weltkoordinatensystems benutzt wird. Zur Vereinfachung wird im folgenden davon ausgegangen, dass dieses Weltkoordinatensystem auch das Weltkoordinatensystem für die graphische Darstellung und für die dreidimensionalen Objekte (Codebookvektoren, Dichteverteilung) darstellt. Dies bedeutet, dass bei einer Bewegung des Handstücks über die Matrix $M_{Phantom}$ die aktuellen Transformationen für die Codebookvektoren ermittelt werden können.

Wird \mathbf{w}'_i als vierdimensionaler Vektor mit einer Eins als vierte Komponente dargestellt, so kann über eine Matrix-Vektormultiplikation der aktuelle \mathbf{w}_i ermittelt werden:

$$\mathbf{w}_i = M_{Phantom} \mathbf{w}'_i + M_{Phantom} \mathbf{R}$$

4.8.2 Kraftherzeugung

Die Kraftrückkopplungsgeräte Phantom der Firma SensAble sind vergleichbar mit einem Roboterarm konzipiert, nur dass der Benutzer dort, wo ein Roboterarm einen Aktuator

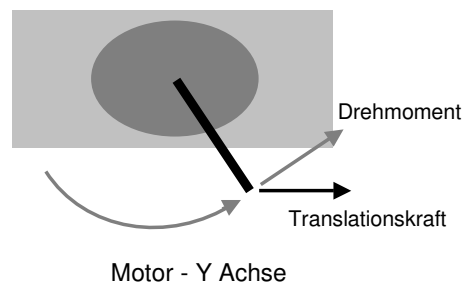


Abbildung 4.16: Drehmoment und Translationskraft

besitzt, einen Stift mit seiner Hand bewegt. Über den Stift kann auf die Hand des Benutzers sowohl ein Drehmoment als auch eine Translationskraft ausgeübt werden.

Die Kräfte werden mit verschiedenen Motoren realisiert, die jeweils ein Drehmoment um eine Achse erzeugen. Das Drehmoment entspricht jedoch nicht einer Translationskraft, sondern ist eine Kraft um eine Drehachse (Abbildung 4.16).

Die Translationskräfte müssen in der Regel, je nach Position der Hand des Benutzers, aus mehreren von den Motoren erzeugten Drehmomenten zusammengesetzt werden. Diese Aufgabe erledigt die Gerätetreibersoftware vollkommen transparent für den Programmentwickler. Die resultierenden Translationskräfte setzen am Anfang des Handstücks an. Da die Translationskraft ein Vektor ist, kann sie leicht mit Hilfe von einer Transformationsmatrix manipuliert werden, zum Beispiel weil die Kraft relativ zu einem anderen Koordinatensystem berechnet worden ist.

In der oben beschriebenen Kraftberechnung wird die Translationskraft relativ zum Schwerpunkt R des Moleküls berechnet. Der Schwerpunkt R wird vom Benutzer mit Hilfe des Handstücks des Phantoms bewegt, so dass die berechneten Kräfte direkt über das Phantom ausgegeben werden können.

Neben den Translationskräften existiert auch das Drehmoment, welches bei dem neuen Phantom 6DOF ebenfalls erzeugt werden kann. Das Drehmoment wird relativ zu der Stiftachse des Handstücks angegeben. Im vorhergehenden Abschnitt wurde das Drehmoment relativ zum Schwerpunkt R des Moleküls berechnet, so dass es ebenfalls ohne weitere Umrechnungen über das Phantom ausgegeben werden kann.

4.8.3 Koordinatensysteme

Bis jetzt wurde davon ausgegangen, dass nur ein Weltkoordinatensystem (welches gleich dem Phantom-Weltkoordinatensystem sein sollte) existiert. Es soll das haptische Rendern jedoch in einem interaktiven Programm zusammen mit einer dreidimensionalen Darstellung umgesetzt werden. Der Benutzer soll hier die Möglichkeit erhalten, sowohl die gesamte Darstellung als auch die einzelnen Objekte (Molekül, Dichteverteilung) zu verschieben und zu drehen.

Um dies zu realisieren, werden weitere (4×4) -Matrizen eingeführt. Es besitzt die Szene eine Matrix $_{World}M_{scene}$, die von dem lokalen Koordinatensystem der Szene in das Weltkoordinatensystem abbildet. Ebenso besitzen die Codebookvektoren eine Matrix $_{Scene}M_{CV}$ und die Dichteverteilung die Matrix $_{Scene}M_{EM}$. Diese Matrizen bilden von den lokalen Objektkoordinatensystemen in das Szenenkoordinatensystem ab (Abbildung 4.17).

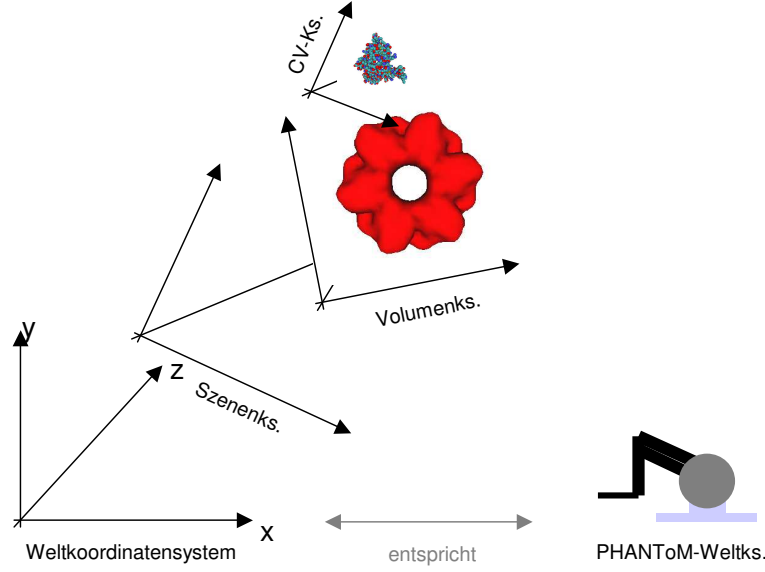


Abbildung 4.17: Koordinatensysteme bei Abbildungsprozess

Will der Benutzer die gesamte Szene drehen, so muss die Matrix $_{World}M_{scene}$ manipuliert werden. Die Objektmatrizen residieren in dem lokalen Szenenkoordinatensystem und müssen daher nicht verändert werden. Soll nur ein einzelnes Objekt verschoben werden, so muss die jeweilige Objektmatrix verändert werden.

Um für die graphische Darstellung die Codebookvektoren und die Dichteverteilung nicht aufwendig umrechnen zu müssen, bietet es sich an, diese schon relativ zu den lokalen Objektkoordinatensystemen anzugeben. Wird ein Objekt verschoben oder die Szene gedreht, so müssen nur die passenden Matrizen manipuliert, aber an den Daten selber nichts verändert werden. Es werden daher Vektoren \mathbf{w}'_{oi} und \mathbf{R}_o eingeführt, welche relativ zu dem Objektkoordinatensystem $_{Scene}M_{CV}$ vorliegen.

Wird das Phantom-Handstück bewegt, so müssen die Codebookvektoren angepasst werden. Das Phantom-Handstück residiert jedoch im Weltkoordinatensystem, so dass die Vektoren zunächst in dieses und nach der Änderung wieder zurück ins Objektsystem transformiert werden müssen:

$$\mathbf{w}'_{oneu_i} = {}_{CV}M_{Scene} {}_{Scene}M_{World} M_{Phantom} {}_{World}M_{Scene} {}_{Scene}M_{CV} \mathbf{w}'_{oi}$$

Für die Kraftberechnung werden die Codebookvektoren im Koordinatensystem der Dichteverteilung benötigt, so dass ebenfalls die Vektoren \mathbf{w}_{ei} und \mathbf{R}_e relativ zu diesem System

eingeführt werden müssen. Sie lassen sich wie folgt berechnen:

$$\mathbf{w}'_{e_i} = {}_{EM}M_{Scene\ Scene}M_{CV}\mathbf{w}'_{o_i}$$

$$\mathbf{R}_e = {}_{EM}M_{Scene\ Scene}M_{CV}\mathbf{R}_o$$

Die Kraft- und Drehmomentvektoren liegen nun aber auch relativ zu diesem Koordinatensystem vor. Für die Kraftausgabe müssen die Matrizen so eingesetzt werden, dass die Kraft- und Drehmomentvektoren wieder relativ zum Weltkoordinatensystem (also dem Phantom-Koordinatensystem) vorliegen:

$$\mathbf{F} = {}_{World}M_{scene\ Scene}M_{EM}\mathbf{F}_e$$

$$\mathbf{N} = {}_{World}M_{scene\ Scene}M_{EM}\mathbf{N}$$

4.9 Zusammenfassung

Es wurde in diesem Kapitel eine Methode beschrieben, mit der ein interaktives Programm den Benutzer mit Hilfe von haptischem Rendern beim Abbildungsvorgang von hochauflösenden Molekülstrukturdaten in niedrigauflösende Dichteverteilungen unterstützen kann.

Der Benutzer soll im vorliegenden Ansatz vom Programm per haptischem Rendern die Information über bessere Abbildungspositionen in der Nähe seiner gerade untersuchten Position erfahren. Dazu wird seine Hand in die Richtung eines ansteigenden Korrelationskoeffizienten gezogen.

Es wurde in diesem Abschnitt die der Kraftberechnung zugrundeliegende physikalisch basierte Modellierung beschrieben. Das Molekül bildet einen starren Körper in einem konservativen Kraftfeld. Das Kraftfeld wird über den Korrelationskoeffizienten als Potentialfunktion definiert.

Ein starrer Körper in einem konservativem System bewegt sich immer in Richtung des nächsten Potentialminimums. Aus dieser Potentialminimierung folgt nach der hier verwendeten Definition des Potentials, dass das Molekül in Richtung des nächsten Korrelationsmaximums strebt. Dies ist genau das bei der Anwendung verfolgte Ziel, soll doch der Anwender durch die Kraftberechnung in die Richtung einer besseren Abbildungsposition gezogen werden. Die Qualität der Abbildungsposition wird dabei über die Höhe des Korrelationskoeffizienten ausgedrückt.

Kapitel 5

Verwandte Arbeiten

Die im vorhergenden Kapitel entwickelte Lösung ist nicht die erste Methode zur Bearbeitung des vorgestellten Abbildungsproblems. Schon seit einigen Jahren ist das Abbilden von hochaufgelösten Molekülstrukturdaten in niedrigaufgelöste Elektronenmikroskopie-Dichteverteilungen ein etabliertes Verfahren.

Die ältesten Methoden sind interaktiver Natur und werden im nächsten Abschnitt beschrieben. Erst in den letzten Jahren wurden auch algorithmische Lösungen entwickelt. Einen kurzen Überblick gibt der übernächste Abschnitt.

5.1 Interaktive Abbildungsmethoden

Die interaktiven Methoden gehen auf eine noch rein mechanische Entwicklung von Fred Richards zurück. Richards hat 1968 den sogenannten “Optical Comparator” [44] entwickelt. Dies ist ein Projektor, in dem mehrere, auf durchscheinende Folien bedruckte Bilder der Dichteverteilung montiert werden. Jedes Bild entspricht einem anderen Schnitt durch das Volumen. Der Projektor erreicht mit einer Spiegeltechnik, dass durch ein projiziertes Abbild des Volumens ein Drahtmodell der Molekülstruktur verschiebbar ist. Danach wird geschätzt, wie gut die Abbildung der hochaufgelösten Struktur in die Dichteverteilung ist. Ist das Ergebnis zufriedenstellend, so werden einfache Messungen mit Hilfe von Linealen und Fadenkreuzen auf der Folie durchgeführt.

Diesen einfachen Projektor entwickelte er weiter und nannte ihn “Richards’ Box”. Die Grundidee des manuellen Verschiebens und Manipulierens der Struktur vor dem Hintergrund der Dichteverteilung blieb jedoch praktisch gleich und hat sich im Grunde auch bei den heutigen Programmen nicht wesentlich geändert. Daher wurden die ersten computerunterstützten Lösungen “Electronic Richards’ Box” genannt.

Als grafikfähige Computer auf den Markt kamen, wurden Programme entwickelt, mit denen Moleküle visualisiert werden konnten. Die meisten dieser frühen Computer waren

nur in der Lage, reine Vektorgraphik darzustellen, da sie direkt den Kathodenstrahl der Bildröhre manipuliert haben. Also wurden zunächst die Bindungen zwischen den Atomen als Vektoren gezeichnet, jedoch die Atome selber nicht besonders hervorgehoben. Für die Dichteverteilungen wurden Oberflächendreiecksnetze erstellt und diese ebenfalls mit Vektoren visualisiert. Diese Darstellung hat sich als übersichtlich erwiesen, so dass auch neuere Programme damit arbeiten.

Frodo

Frodo [29] ist ein interaktives Programm, das von Alwyn Jones in den 1970er Jahren am Max-Planck-Institut in München entwickelt worden ist. Es lief zunächst auf einer DEC PDP 11/40 und einem Vector General 3404 Display System. Das Programm konnte per Stift und Graphiktablett bedient werden und hat die Moleküle als einfache Strichmodelle dargestellt. Der Benutzer konnte sowohl Atome als auch ganze Teilbereiche des Moleküls bewegen und so die Abbildung in die niedrigauflösenden Daten herstellen.

Später sind weitere Versionen auf neueren Computern entstanden, die zum Teil von Jones (wie die Version für die Evans and Sutherland Picture System 2), zum Teil aber von anderen Gruppen entwickelt worden sind. Frodo hat eine große Verbreitung gefunden und war in den achtziger Jahren die am häufigsten eingesetzte Eletronic Richards' Box.

O

Später hat Jones ein neues Programmsystem erstellt. Dieses Programm nannte er "O" [28], und es hat in der Folgezeit Frodo weitestgehend verdrängt. O hat eine erweiterte graphische Darstellung (Abbildung 5.1), vor allem wurden aber die Interaktionsmöglichkeiten verbessert. Der Benutzer kann eine heute übliche Computermaus einsetzen. Daneben haben die sogenannten "Dial Boxes" eine weite Verbreitung gefunden. Eine Dial Box besteht aus einer Menge von analogen Drehreglern und digitalen Tastern, die das Programm für verschiedene Funktionen einsetzen kann. Es ist möglich, Objekte mit Hilfe der Regler zu verschieben und um Punkte zu rotieren. Eine Dial Box besitzt neben der höheren Anzahl an Freiheitsgraden zudem eine größere Präzision verglichen mit einer normalen Computermaus.

O besitzt eine sehr flexible Benutzeroberfläche, die von den Benutzern leicht auf die persönlichen Vorlieben angepasst werden kann. Es existiert eine Kommandozeile, in der der Benutzer Befehle an das Programm absetzen kann. Diese Befehle können in Makros zusammengefasst werden. Auch die im Programm angezeigten Menüs lösen diese Befehle aus. Der Benutzer kann die Menüs umgestalten und dort Makros oder Befehlsketten einsetzen. Gerade häufig benötigte Abläufe können so wesentlich beschleunigt und die manuell zu verrichtende Arbeit minimiert werden.

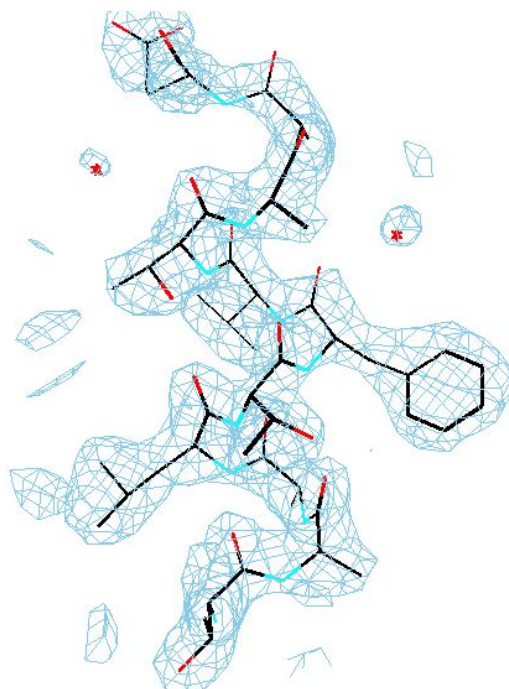


Abbildung 5.1: Das interaktive Einpassprogramm „O“

Da O bereits aus dem Jahre 1991 stammt, ist es aus der Sicht des Benutzeroberflächendesigns und der graphischen Möglichkeiten heute überholt. Es wird von seinem Autor gepflegt, wenn auch größere Änderungen an dem Programm in den letzten Jahren ausgeblieben sind. Trotzdem ist es nach wie vor das am meisten genutzte Programm, um einen interaktiven Einpassprozess durchzuführen.

5.2 Algorithmische Abbildungsmethoden

Die in Kapitel 4 vorgestellten Grundlagen bilden nicht nur die Grundlagen für die in dieser Arbeit entwickelte Lösung, sondern werden auch von den meisten algorithmischen Ansätzen verwendet.

Viele Programme verwenden einen ähnlichen Ansatz wie der im vorhergenden Kapitel vorgestellte einfache Lösungsvorschlag (siehe Kapitel 4). Zur Bewertung einer Abbildungssituation wird ebenfalls meist ein Korrelationskoeffizient verwendet, so dass sich eine vergleichbare Programmstruktur ergibt.

Zunächst wird eine PDB-Datei für die Molekülstruktur sowie eine Dichteverteilung eingelesen. Danach wird die Struktur in der Auflösung an die Dichteverteilung angepasst, und so ein zweiter Volumendatensatz erzeugt.

Beide Volumendatensätze werden in einer Hauptschleife der Algorithmen mit Hilfe eines Korrelationskoeffizienten verglichen. Im Gegensatz zur interaktiven Lösung werden alle

möglichen Positionen und dort jeweils alle möglichen Orientierungen auf die Korrelation der Datensätze hin überprüft. Ein maximaler Korrelationskoeffizient wird so gesucht und am Ende zusammen mit der passenden Position und Orientierung ausgegeben.

Es ergibt sich ein in Abbildung 5.2 dargestellter Programmablauf.

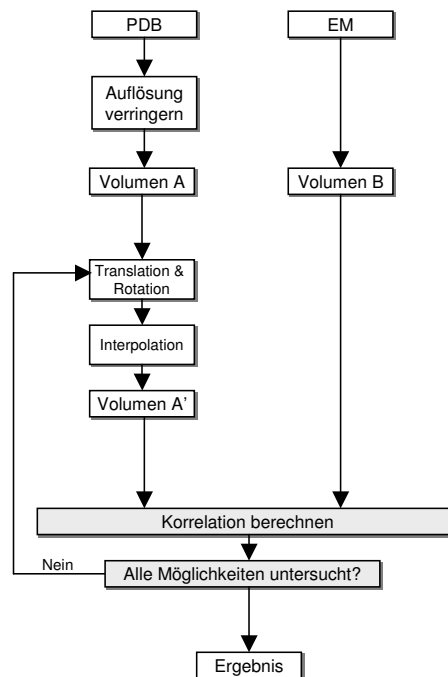


Abbildung 5.2: Programmablauf einer algorithmischen Lösung

Das Problem eines solchen Ansatzes sind die durch den großen Suchraum bedingten langen Programmlaufzeiten. Die Größe des Suchraumes ergibt sich, da die Korrelation an jeder Position für jede mögliche Rotation berechnet werden muss. Zudem ist jede Korrelationsberechnung aufwendig, müssen doch die beiden kompletten Volumendatensätze untersucht werden. Bei einem lokal normierten Korrelationskoeffizienten muß zudem die Standardabweichung des interpolierten Datensatzes immer wieder neu berechnet werden.

Ein Beispiel für die Implementierung eines solchen Ansatzes ist das Programm COAN [60] von Volkmann, das einen einfachen Kreuzkorrelationskoeffizienten über dem gesamten Suchraum berechnet. Neuerdings wurde das Programm durch einen Segmentationsalgorithmus [59] erweitert. Volkmann sieht darin die Möglichkeit, sein Programm auch auf extrem niedrig aufgelöste Tomographiedaten anwenden zu können.

Die Anwendung der in Kapitel 4 vorgestellten Korrelationskoeffizienten ist nicht unproblematisch. Es zeigt sich, dass meist eine größere Menge an potentiellen Maxima des Korrelationskoeffizienten vorhanden ist. Da sich die Maxima oft numerisch nur wenig unterscheiden, ist eine Auswahl des globalen Maximums und somit der besten Abbildungsposition ausschliesslich auf der Grundlage dieses Koeffizienten problematisch.

Der einfache unnormierte Korrelationskoeffizient wird bei einer Anwendung auf die vorliegende Problemstellung zu:

$$C(R, T) = \int \rho_{em}(r) \rho_{calc}(r, R, T) d^3r$$

Wird dieser Koeffizient als Bewertungsfunktion in einem Algorithmus eingesetzt, so wird das Programm die beste Abbildungsposition an einer Stelle erkennen, an der die Dichte ρ_{em} besonders hoch ist. Soll eine Teilkomponente in eine größere Dichteverteilung abgebildet werden, so wird dies dazu führen, dass bei geringeren Auflösungen (zirka 10 Ångström und schlechter) die beste Abbildungsposition immer in der Mitte der Verteilung gefunden wird. Bei solch geringen Auflösungen besitzen die Elektronenmikroskopiedaten praktisch keine innere Struktur mehr, so dass die Überlappung der beiden Datensätze und somit der Term $\rho_{em}(r) \rho_{calc}(r, R, T)$ im Zentrum maximal wird.

Roseman hat ein Programm DOCKEM [45] entwickelt, welches mit Hilfe einer lokal normierten Kreuzvarianzfunktion die Korrelation berechnet und damit auch den gesamten Suchraum absucht. Um das Streben des Korrelationskoeffizienten in Richtung des Zentrums zu verhindern, wird die Korrelation nur für die Teile der Datensätze berechnet, die sich überlappen:

$$C(R, T) = \frac{\int_{mask} \rho_{em}(r) \rho_{calc}(r, R, T) d^3r}{\sqrt{\int_{mask} \rho_{em}^2(r) d^3r} \sqrt{\int_{mask} \rho_{calc}^2(r) d^3r}}$$

Dabei entspricht *mask* einer Maskierung, die die gemeinsamen Voxel herausfiltert. Durch die Normierung des Koeffizienten wird für die Positionen innerhalb des Elektronenmikroskopiedatensatzes nicht mehr automatisch eine höhere Korrelation berechnet. Durch die Anwendung des komplexeren Korrelationskoeffizienten ist die Berechnung aber sehr aufwendig.

Durch diese und andere Nachteile des Korrelationskoeffizienten motiviert, existieren verschiedene Ansätze, bessere Bewertungsfunktionen für eine Abbildung zu entwickeln (einen Überblick gibt [66]). Eine universell anwendbare Bewertungsfunktion, die nicht zu falsch-positiven Resultaten oder zu langen Rechenzeiten führt, konnte aber noch nicht gefunden werden.

Neben diesen Entwicklungen, die mehr oder weniger den oben skizzierten Standardalgorithmus implementieren, existieren auch andere Ansätze. Zwei Programme werden im folgenden kurz vorgestellt. ESSENS stellt eine der ältesten algorithmischen Lösungen dar, das Programm SITUS hingegen ist wegen seiner alternativen Algorithmusidee hervorzuheben.

ESSENS

Eines der ersten algorithmischen Programme war ESSENS von Kleywegt und Jones [31]. Ihr Ansatz ist ein Suchalgorithmus, der das Molekül in die Volumendaten mit allen mög-

lichen Translationen und Rotationen abbildet. Für jede der möglichen Positionen rechnet der Algorithmus für jedes Atom des Moleküls einen Durchschnittsdichtewert aus. Hierzu bildet er den Mittelwert aller acht um das Atom herum liegenden Dichtewerte.

Diese Mittelwerte aller Atome des Moleküls werden sortiert und die K kleinsten Werte werden aufsummiert und durch K geteilt. Die Idee hierbei ist, dass ein hoher Dichtewert ein positives Abbildungskriterium für ein Atom ist. Für das gesamte Molekül gilt, dass bei einer guten Abbildungsposition auch die schlechtesten Atome noch recht gut abschneiden sollten. Stimmt nur ein Teil des Moleküls an der aktuellen Position mit dem Volumendatensatz überein, so ist die Abbildungsposition negativ zu bewerten.

ESSENS bildet aus den Ergebnissen einen neuen Volumendatensatz, bei dem an jeder Position der beste Wert von allen Rotationen an dieser Stelle verzeichnet wird. Dieser Volumendatensatz kann mit dem Programm O (siehe vorhergehenden Abschnitt) angezeigt werden, der Anwender bekommt so eine Idee von der besten Abbildungsposition.

Der Nachteil dieses Ansatzes ist der enorme Rechenzeitbedarf, der bei größeren Datensätzen wegen dem immer weiter steigenden Suchraum schnell dazu führt, dass das Programm nicht mehr einsetzbar ist. Gerade die aufwendigen Berechnungen an jeder Suchposition inklusive des Sortiervorgangs treiben die Rechenzeit in die Höhe. Die Autoren schreiben hierzu, dass der Algorithmus durch das Sortieren unempfindlicher gegenüber dem Rauschen in den Datensätzen wird.

Die hohen Rechenzeitanforderungen haben Cowtan dazu gebracht, eine beschleunigte Version des Algorithmus zu entwickeln [15]. Er führt die Suche bezüglich der Translationen im Fourierraum durch, was eine erhebliche Beschleunigung bewirkt.

Situs

Situs [68] ist eines der wenigen Programmpakete, welches sich innovativ mit der Suchraumproblematik auseinandergesetzt hat. Die beiden Datensätze werden hier durch Vektorquantisierung stark vereinfacht, so dass sich ein wesentlich kleinerer Suchraum ergibt.

Situs besteht aus einer Reihe von Anwendungen, welche jeweils Teilschritte des gesamten Einpassprozesses übernehmen (Abbildung 5.3).

Der erste Schritt besteht in einer Konvertierung der Eingabedaten in ein eigenes Situs-Dateiformat (Teilprogramm CONVERT). Dieses Dateiformat kann danach sehr leicht von den anderen Teilprogrammen eingelesen und bearbeitet werden. Bei den Molekülsstrukturdaten existiert mit dem PDB-Format [8] ein weit verbreitetes Standardformat. Situs benutzt daher durchgängig dieses Format und implementiert kein eigenes.

Mit einigen weiteren Programmen können die Elektronenmikroskopie-Dichteverteilungen analysiert und bearbeitet werden. Zum Beispiel ist es möglich, mit SUBTRACT zwei Verteilungen voneinander zu subtrahieren, mit INVERT das Vorzeichen aller Werte zu ändern oder mit HISTOVOX ein Histogramm der Werte anzuzeigen.

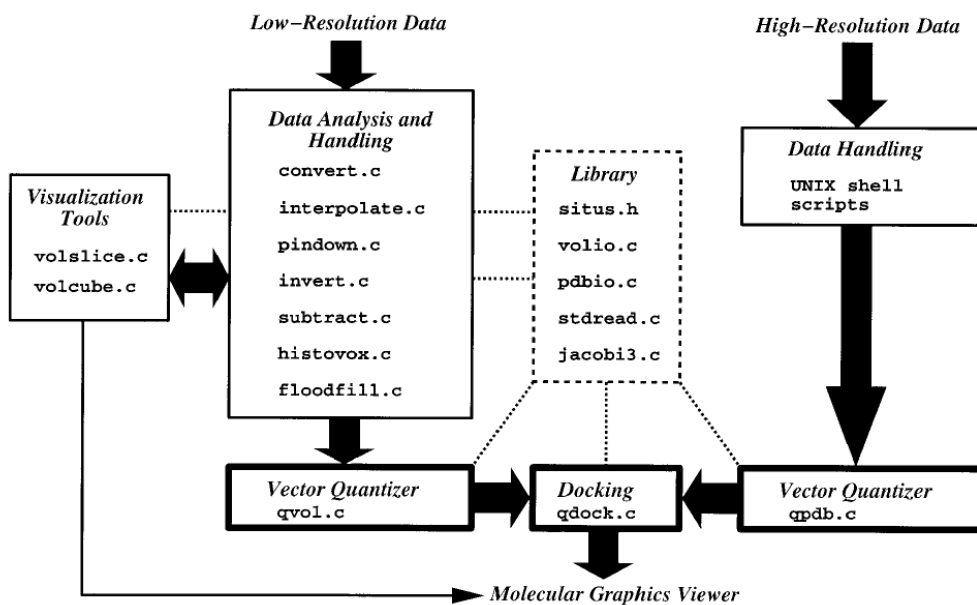


Abbildung 5.3: Übersicht über das Situs Programmpaket

Die Grundidee des Abbildungsalgorithmus besteht zunächst darin, die Datensätze zu vereinfachen. Situs benutzt hierzu die Technik des Vektorquantisierens, indem es sowohl den Linde-Buzo-Gray [34] als auch den Topology-Representing-Networks-Algorithmus [36] implementiert (siehe hierzu auch Kapitel 4.5). Die Algorithmen werden genutzt, um sowohl die hochaufgelösten Molekülstrukturdaten als auch die niedrigaufgelösten Datensätze durch eine kleine Gruppe von sogenannten Codebookvektoren zu ersetzen [69]. Die Teilprogramme heissen QPDB beziehungsweise QVOL.

Nachdem beide Datensätze durch eine Menge von k Codebookvektoren beschrieben werden können, vereinfacht sich der eigentliche Einpassprozess. Der hochaufgelöste PDB-Datensatz sei durch die Vektoren $x(i)$ beschrieben, und der niedrigaufgelöste Datensatz durch $y(j)$ mit $i, j = 1, \dots, k$. Gesucht ist eine Abbildung $I : j \rightarrow i$, die jedem Vektor $y(j)$ einen Vektor $x(i)$ zuordnet. Durch diese Abbildung ist bekannt, welcher Codebookvektor aus dem hochaufgelösten Datensatz welchem Vektor aus dem niedrigaufgelösten Datensatz entspricht. Es sind nur noch die Vektoren $x(i)$ auf die Vektoren $y(j)$ aufeinander zuzubewegen, so dass die Abstände insgesamt minimal werden.

Leider ist diese Abbildung I zunächst unbekannt. Um diese Abbildung zu ermitteln, führt das Programm QDOCK eine vollständige Suche durch. Jede Abbildung I kann durch eine Permutation beschrieben werden. QDOCK untersucht daher alle möglichen Permutationen. Für jede Permutation bestimmt QDOCK die Standardabweichung zwischen $x(i)$ und $y(j)$, nachdem sie mit minimalem Abstand zueinander positioniert worden sind. QDOCK sortiert die Permutationen je nach zugehörigem Wert der Standardabweichung in eine Tabelle ein und berechnet für die höchsten 20 Ergebnisse den Korrelationskoeffizienten. Dazu werden

jedoch die Ursprungsdatensätze analog zu den Codebookvektoren positioniert und dann der Korrelationskoeffizient berechnet. In der Regel ergibt sich analog zu den Werten der Standardabweichung auch die Reihenfolge in den Korrelationskoeffizienten (ein niedriger Wert ergibt einen hohen Korrelationskoeffizienten). Der Tabellenerste ist somit die gesuchte Lösung des Abbildungsproblems.

Die Stärke in dem Ansatz liegt klar in der drastischen Verkleinerung des Suchraumes. In Situs wird das Problem auf das Suchen der richtigen Permutation I zurückgeführt, was selbst bei der hier durchgeführten Breitensuche nur $k!$ Möglichkeiten ergibt (wobei k die Anzahl der Codebookvektoren ist). Da k zumeist ein Wert im Bereich 5 bis 15 ist, ist der Suchraum vergleichsweise klein. Zudem kosten die jeweils durchzuführenden Berechnungen durch die geringe Anzahl an Codebookvektoren wesentlich weniger Rechenzeit als die Berechnung des Korrelationskoeffizienten auf den kompletten Ursprungsdaten.

Der Einsatzzweck des Algorithmus ist das Abbilden von kompletten Molekülen. Ein Abbilden von Teilmolekülen in große Dichteverteilungen ist so nicht möglich, da hier die 1 : 1 Beziehung zwischen Codebookvektoren des Moleküls und der Verteilung nicht mehr gilt. Für solche Anwendungen wurde von Wriggers und Chacón [13] ein neues Programm mit dem Namen COLORES entwickelt, welches ebenfalls Teil des SITUS-Programmpaketes ist. Es implementiert eine vollständige Suche mit Hilfe eines Laplace-Filters, welcher die Konturen innerhalb der Dichteverteilung besonders betont. Dadurch wird das Programm auch bei sehr niedrigen Auflösungen einsetzbar. Durch eine Optimierung mit Hilfe von Fourierraummethoden kann zudem die Ausführungsgeschwindigkeit trotzdem mit anderen Programmen konkurrieren.

5.3 Vergleich interaktiver und algorithmischer Methoden

Die interaktiven Programme sind verglichen mit den algorithmischen Ansätzen sehr populär. Dies ist zunächst verwunderlich, geben die algorithmischen Lösungen doch vor, das gesamte Abbildungsproblem vollautomatisch bearbeiten zu können. Beim konkreten Einsatz einer solchen Lösung ergeben sich jedoch mehrere Probleme.

Ein Problem ist der hohe Rechenzeitbedarf der Programme. Bedingt durch den sehr großen Suchraum und die komplexen Berechnungen für jede mögliche Abbildung, steigt der Rechenzeitbedarf bei größeren Aufgabenstellungen schnell an.

Ausserdem stammen die verwendeten Datensätze aus Experimenten und sind daher nicht frei von verschiedenen Defekten. Ein hoher Rauschpegel, wenig Kontrast oder fehlerhafte Bereiche sind typische Eigenschaften der Daten. Die algorithmischen Programme sind anfällig gegenüber diesen Störungen, da sie die Information nur auf der Grundlage der Korrelationskoeffizienten bewerten können. Gerade die einfacheren und wegen ihrer vergleichsweise schnellen Berechnung beliebten Koeffizienten sind anfällig und erkennen gu-

te Abbildungspositionen an Stellen, die für einen menschlichen Betrachter offensichtlich falsch sind.

Die eingesetzten Korrelationskoeffizienten sind als Bewertungskriterium eine weitere Fehlerquelle. Ihre Möglichkeiten, die Eigenschaften der Daten zu vergleichen, sind sehr begrenzt. Daraus resultieren diverse Probleme, die in den vorhergehenden Abschnitten angedeutet worden sind.

Ein weiteres Problem ist das eingesetzte “Rigid-Body Docking”. Da beide Datensätze aus verschiedenen Experimenten stammen, unterscheiden sie sich nicht nur in der Auflösung. Teile des Moleküls haben eine unterschiedliche Lage relativ zum Gesamtojekt, so dass das “Rigid-Body Docking” gar keine perfekte Lösung finden kann. Ein “Flexible Fitting” würde jedoch den Suchraum noch wesentlich vergrößern, so dass dies meist nicht in Frage kommt.

Insgesamt kann festgestellt werden, dass beide Ansätze Schwächen aufweisen. Ein vollautomatisch perfekt arbeitender Algorithmus wäre wünschenswert, aber aus verschiedenen Gründen existiert ein solcher nicht. Für einige Datensätze stellen die Algorithmen schon sehr gute Lösungen dar, für andere scheiden sie aber auch aus. Die Alternative der interaktiven Programme schiebt die Lösung auf den Experten ab. Gerade bei großen Systemen und bei fehlerhaften Mikroskopiedatensätzen ist die Komplexität der Aufgabenstellung auch für den Experten sehr groß.

Aus diesen Gründen werden sowohl interaktive als auch vollautomatische Programme angewendet. Je nach Einsatzgebiet muss entschieden werden, welcher Ansatz der überlegene ist und die präzisesten Ergebnisse liefert. Aufgrund dieser Erfahrungen schlägt die vorliegende Arbeit einen interaktiven Ansatz vor, in dem Methoden der algorithmischen Lösungen angewandt werden. Der Experte wird bei der Arbeit durch das haptische Rendern unterstützt, ohne seine Entscheidungsfreiheit bezüglich einer günstigen Abbildungsposition zu verlieren. Er erhält durch die Kraftausgabe nur eine zusätzliche Information, die ihm anzeigt, wie ein Algorithmus die aktuelle Abbildung bewerten würde.

Da die Kraftausgabe nur eine zusätzliche Hilfestellung darstellt, können diverse Einschränkungen der algorithmischen Lösungen vermieden werden. Falsch-positive Ergebnisse können von dem Experten in einem interaktiven Prozess unmittelbar erkannt und ignoriert werden. Das Problem der vielen lokalen Maxima, die sich numerisch nur wenig unterscheiden, spielt in einem interaktiven Prozess ebenfalls keine Rolle. Der Experte bewertet die Situation anhand der zusätzlichen visuellen Information und wählt auf der Grundlage seines Kontextwissens ein Maximum, also eine Position mit minimaler Kraftausgabe, als beste Abbildungsposition aus.

5.4 Anwendungsgebiete

Für die hier beschriebenen Abbildungsverfahren existieren mehrere Anwendungsgebiete. Wie schon beschrieben, können verschiedene Strukturermittlungsverfahren eingesetzt werden. Bei jeder Mikroskopieaufnahme stellt sich die Frage, wie diese zu interpretieren ist, ob wirklich das gesuchte Molekül abgebildet werden konnte. Im Rahmen einer solchen Interpretation ist es nützlich, wenn das Abbildungsverfahren angewendet werden kann und hochaufgelöste Strukturen in die neuen Volumendaten positioniert werden können. So wird eine Interpretation einfacher und für andere leichter nachvollziehbar.

Eine andere Anwendungsmöglichkeit liegt darin, dass keines der hochaufgelösten Verfahren beliebig große Strukturen in atomarer Auflösung untersuchen kann. Sind große Strukturen zu untersuchen, so müssen in der Regel niedrigaufgelöste Verfahren angewendet werden. Um doch noch eine Aufnahme in atomarer Auflösung zu erhalten, kann ein Abbildungsverfahren zum Einsatz kommen. Teilbausteine des großen Moleküls müssen mit Hilfe von hochaufgelösten Verfahren gewonnen werden, damit diese dann in die niedrigaufgelösten Daten abgebildet werden können.

Eine weitere Anwendungsmöglichkeit beschreibt Rossman in [46]. Selbst wenn die komplette Struktur mit Hilfe von Roentgenkristallographie auf nahezu atomarer Auflösung abgebildet werden kann, so ist ein per Einpassprozess entstandenes Modell trotzdem sinnvoll. Niedrigaufgelöste Elektronenmikroskopiedaten können mit relativ geringem Präparationsaufwand gewonnen werden. Mit ihrer Hilfe und mit Hilfe der Abbildungsverfahren kann schon vor der Roentgenkristallographie ein vorläufiges Modell der Struktur erstellt werden. Dies kann den eigentlichen Prozess der Roentgenkristallographie erleichtern und zudem können die später gewonnenen Daten hiermit verbessert und Fehler gefunden werden.

Kapitel 6

SenSitus

In diesem Kapitel wird die Realisierung des in Kapitel 4.6 vorgestellten Programmentwurfs beschrieben. Umgesetzt wurde der Entwurf in Form eines Programmes mit dem Namen “SenSitus”.

SenSitus besteht aus mehreren Komponenten, die aus Gründen der Wiederverwendbarkeit voneinander getrennt entwickelt worden sind. So ist es nach Abschluss dieser Arbeit möglich, Teilkomponenten in späteren Projekten einzusetzen.

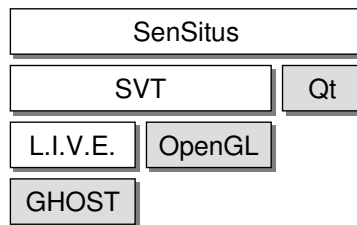


Abbildung 6.1: Übersicht über die Architektur von SenSitus

Die in Abbildung 6.1 dargestellten Komponenten werden in den folgenden Abschnitten dieses Kapitels beschrieben. SenSitus verwendet zur Implementierung der Visualisierung das “Scientific Visualization Toolkit” (kurz SVT), welches im Abschnitt 6.2 besprochen werden soll. Die Ansteuerung der Kraftrückkopplungsgeräte wird von der Bibliothek L.I.V.E. übernommen, welche genau wie SVT im Rahmen dieser Arbeit entstanden ist, und in Abschnitt 6.1 besprochen wird.

Für die Realisierung der graphischen Benutzerschnittstelle wird aus Gründen der Portabilität die Bibliothek Qt verwendet. Diese Bibliothek ist ein kommerzielles Produkt der Firma Trolltech[57], steht jedoch für Unix Betriebssysteme kostenlos zur Verfügung. Qt ist darüberhinaus für Microsoft Windows und Apple Macintosh erhältlich.

Im Anschluss an die Beschreibung von L.I.V.E. und SVT wird genauer auf das eigentliche Hauptprogramm SenSitus eingegangen.

6.1 L.I.V.E.

Virtual-Reality-Systeme benötigen viele Dateneingabegeräte. Oft wird ein sogenanntes Headtracking-System zur Positionsbestimmung (siehe Kapitel 2.2.1), oder dreidimensionale Eingabegeräte wie ein “Wand” zur Interaktion eingesetzt. Darüberhinaus ist die Mensch-Maschine-Interaktion ein wichtiger VR-Forschungsschwerpunkt, bei dem die Eingabegeräte eine bedeutende Rolle spielen. Das Entwickeln von Spezialgeräten ist daher oft zwingend notwendig.

Durch diese Vielfalt existieren diverse Programmierschnittstellen, mit deren Hilfe die Geräte angesprochen werden können. Für den Entwickler eines Anwendungsprogrammes ist diese Vielfalt problematisch, muss er doch alle Schnittstellen in seinem Programm unterschiedlich ansprechen.

Es existieren einige Softwarebibliotheken, die eine allgemeine Schnittstelle für VR-Geräte anbieten. Ein Anwendungsprogramm kann durch diese Bibliotheken auf eine einheitliche Schnittstelle für alle Eingabegeräte zugreifen. Die meisten dieser Bibliotheken sind kommerzielle Produkte, die durch ihren hohen Kaufpreis den Einsatz im Rahmen dieser Arbeit verhindert haben. Die wenigen kostenlosen Softwarebibliotheken haben diverse Einschränkungen (eingeschränkte Portabilität, keine Treiber für am Zentralinstitut für Angewandte Mathematik genutzte Geräte, begrenzte Stabilität), so dass ihr Einsatz ebenfalls verworfen werden musste.

Aus diesen Gründen wurde eine Bibliothek entwickelt, die die Ansteuerung der VR-Geräte übernimmt. Der Name L.I.V.E. steht für “Library for Input Devices in VR-Environments” und beschreibt somit den Einsatzzweck.

6.1.1 Aufbau von L.I.V.E.

Fast alle heute verfügbaren Bibliotheken teilen die unterschiedlichen VR-Geräte in Geräteklassen ein. VRPN führt zum Beispiel die Kategorien “Tracker”, “Button” oder auch “Kraftrückkopplungsgeräte” ein (es existieren noch wesentlich mehr Kategorien).

Die zugrundeliegende Idee ist, dass der Anwendungsprogrammierer sich nicht mehr mit den spezifischen Eigenschaften eines konkreten Gerätes beschäftigen muss. Er programmiert die generische Gerätekategorie und erst beim konkreten Einsatz wird über den eigentlichen Gerätetyp entschieden.

Eine solche Einteilung der Geräte ist nicht unproblematisch. Bei einer großen Anzahl von Kategorien mit sehr geringen Unterscheidungsmerkmalen ist die Anzahl an Geräten, die in eine Kategorie eingeteilt werden, gering. Ein Anwendungsprogramm kann ohne umfangreiche Änderungen am Quelltext nur wenige Geräte nutzen. Weiterhin ist es dann unwahrscheinlich, dass sich ein neues Eingabegerät in eine der bestehenden Kategorien einteilen

läßt. Oft müssen neue Kategorien eingeführt werden, die dann zunächst von keinem bestehenden Anwendungsprogramm genutzt werden.

Geräte können Eigenschaften besitzen, die eine Einteilung in mehrere Kategorien sinnvoll machen. Das in dieser Arbeit eingesetzte Krafrückkopplungsgerät Phantom (siehe Abschnitt 3.3) kann sowohl zur Kraftausgabe als auch nur zur Positionsbestimmung der Hand eingesetzt werden. Wird es nur für die Positionsbestimmung eingesetzt, dann könnte es ein Tracking-System ersetzen. Eine mehrfache Einteilung der Geräte wird von den Bibliotheken aber oft nicht unterstützt.

Aus diesen Gründen wurde eine Einteilung in Kategorien bei L.I.V.E. nicht vorgesehen. L.I.V.E. beschränkt sich auf die Ansteuerung der Geräte und den notwendigen Datenaustausch. Jedes Gerät kann Daten verschiedenen Typs zur Verfügung stellen oder empfangen. Die Interpretation der Daten wird dem Anwendungsprogramm beziehungsweise dem Gerät überlassen.

6.1.2 Datenkanäle

Die Daten werden dem Anwendungsprogramm in Form von Datenkanälen übergeben, die das Eingabegerät in beliebiger Anzahl erzeugen kann. Jeder Kanal stellt genau einen Datenstrom eines bestimmten Typs und einer bestimmten Richtung (von oder zu dem Eingabegerät) dar (Abbildung 6.2).

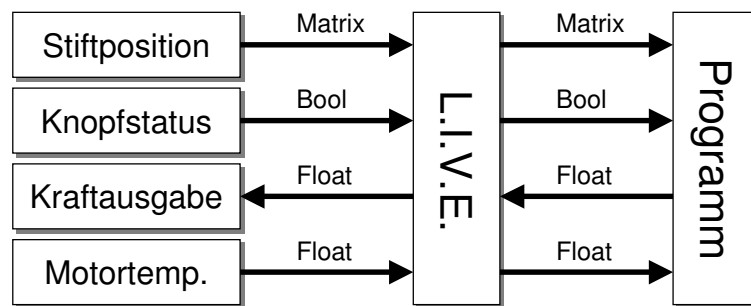


Abbildung 6.2: Datenkanäle am Beispiel des Phantom

Ohne die Festlegung, wie die Daten zu verwenden sind, können die Datenkanäle an beliebigen Punkten im Programm eingesetzt werden. Der Knopf des Phantom-Stiftes muss nicht unbedingt etwas mit Krafrückkopplung zu tun haben, kann zum Beispiel auch eine Lichtquelle einschalten. Mit der Stiftposition kann die Perspektive manipuliert und die Kraftausgabe überhaupt nicht verwendet werden.

Nicht nur kommerzielle Geräte sondern auch das am Zentralinstitut für Angewandte Mathematik entwickelte Eingabegerätesystem “AD-Box” [5] konnte in das L.I.V.E.-System integriert werden. Die AD-Box stellt eine Analog-Digital-Wandlereinheit dar, die über 4 analoge und 4 digitale Eingänge verfügt. Die Eingänge werden abgefragt und als digitale Werte per serieller Schnittstelle an den Computer übermittelt.

Die AD-Box eignet sich vor allem für die Entwicklung von Eingabegeräten. Über die analogen und digitalen Eingänge können verschiedene Potentiometer und Taster abgefragt werden. Eine genaue Einordnung eines solchen Universalgerätes in eine Kategorie wäre schwierig. Im L.I.V.E.-System stellt jeder Eingang einen Kanal dar und kann sofort verwendet werden.

Ein weiterer Vorteil dieses Datenkanalkonzeptes ist die geringe Komplexität. Die Kategorien der anderen Bibliotheken werden zumeist auf C++ Klassen abgebildet. So entsteht eine große Klassenstruktur, die nicht nur anspruchsvoll zu entwickeln ist, sondern für den Anwendungsprogrammierer auch Einarbeitungsaufwand erfordert. Im Gegensatz dazu ist das Datenkanalkonzept von L.I.V.E. einfach zu erlernen und führt zu einer sehr kompakten Bibliothek mit nur wenigen Klassen.

6.1.3 Treibermodule

Die meisten Bibliotheken zur Ansteuerung von VR-Geräten besitzen eine monolithische Struktur. Routinen zum Ansprechen der Geräte werden meist in die Bibliothek integriert. Problematisch ist die immer weiter anwachsende Größe der Bibliothek und die zunehmende Abhängigkeit von diversen Zusatzbibliotheken der Hersteller der VR-Geräte.

Im Falle von L.I.V.E. wurde ein anderer Ansatz verfolgt. In der Grundbibliothek, mit der die Anwendungen linken, wurden keine Ansteuererroutinen für die Eingabegeräte implementiert. Diese werden erst zur Laufzeit des Anwendungsprogramms von L.I.V.E. nachgeladen.

Die Ansteuererroutinen werden dazu in Treibermodule ausgelagert. Jedes Treibermodul muß dazu über einen klar definierten Satz von Funktionen verfügen. Beim Nachladen des Moduls überprüft L.I.V.E. das Vorhandensein dieser Funktionen und benutzt diese, um das Gerät zu initialisieren und Daten mit ihm auszutauschen. Durch das Datenkanalkonzept war nur eine geringe Anzahl an Funktionen notwendig.

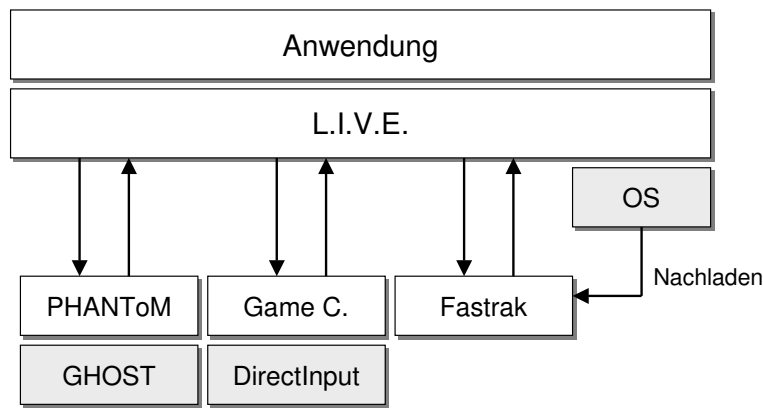


Abbildung 6.3: L.I.V.E.-Treibermodule

Durch das Nachladen der Treibermodule ergeben sich vielfältige Vorteile. Der Anwendungsprogrammierer muss zur Compilezeit noch nichts über das konkrete Eingabegerät wissen, welches der Benutzer später zur Laufzeit einsetzen möchte. Das Programm wird nur an die L.I.V.E.-Bibliothek gebunden, bleibt aber unabhängig von dem konkreten Treibermodul. Der Anwendungsprogrammierer kann so sein Programm mit den ihm zur Verfügung stehenden Geräten testen, der Endanwender kann später ganz andere Geräte einsetzen.

Darüber hinaus ist es so möglich, neue Eingabegeräte zu nutzen, die zum Zeitpunkt des Erstellens der Anwendung noch unbekannt sind. Die neugeschriebenen Treibermodule lassen sich trotzdem mit dem Programm einsetzen — ein Neuübersetzen der Anwendung ist nicht notwendig.

Ebenfalls wird die Portierung der Bibliothek auf verschiedene Betriebssysteme wesentlich erleichtert. Es muss nur die Grundbibliothek portiert werden; es ist nicht notwendig, alle Treibermodule auf ein neues Betriebssystem zu übertragen. Für einige Geräte stehen die Ansteuerrouninen oder Bibliotheken des Herstellers nur für bestimmte Betriebssysteme zur Verfügung. Durch das Treibermodulkonzept bleibt L.I.V.E. trotzdem portabel.

Durch die Austauschbarkeit der Module kann zudem die Entwicklung leicht dezentralisiert werden.

Zur Zeit existieren Treibermodule für die am Zentralinstitut für Angewandte Mathematik existierende Hardware. Folgende Treibermodule wurden realisiert:

- Polhemus Fastrak
- SensAble Phantom (einschliesslich Unterstützung 6DOF Version)
- Logitech Spacemouse
- AD-Box
- PC Eingabegeräte über Ansteuerung der Microsoft DirectInput Bibliothek (inklusive Unterstützung der Krafrückkopplung)

6.1.4 Netzwerkzugriff

Mit Hilfe von L.I.V.E. kann auf ein Eingabegerät über das Netzwerk zugegriffen werden. Dies war in der vorliegenden Anwendung notwendig, da das verwendete Krafrückkopplungsgerät Phantom nur unter dem Betriebssystem Windows einsetzbar war. Gleichzeitig wird das verwendete VR-System von einem SGI Graphikcomputer unter dem Betriebssystem Irix betrieben. Eine direkte Ankopplung war somit nicht möglich.

Grundsätzlich ist ein solcher Netzwerkzugriff aber auch dann sinnvoll, wenn die Ansteuerung des Gerätes viel Rechenleistung benötigt. Der Graphikcomputer würde unter diesen

Umständen bei einer direkten Kopplung Rechenleistung verlieren, die für die Graphikausgabe nicht mehr zur Verfügung stehen würde.

Durch das oben beschriebene Modulkonzept läßt sich der Netzwerkzugriff elegant realisieren. Es wurde ein neues Treibermodul “Netzwerk Client” und eine Applikation “Netzwerk Server” entwickelt, die sich beide ohne Änderung der Hauptbibliothek in das L.I.V.E.-Konzept integrieren liessen.

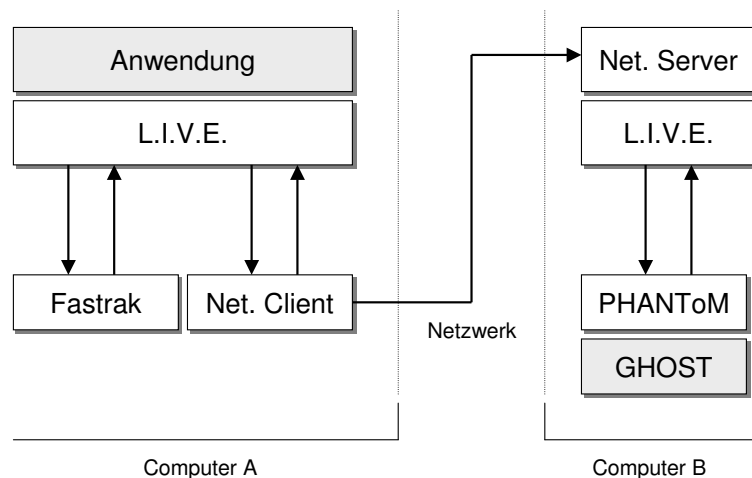


Abbildung 6.4: L.I.V.E.-Netzwerkzugriff

Das Treibermodul wird anstelle des normalen Treibermoduls in der Anwendung eingesetzt. Es ermittelt alle Aufrufe der Applikation an das Gerät und überträgt sie über das Netzwerk an die Serverapplikation. Diese wiederum ist eine normale L.I.V.E.-Anwendung, kann also ebenfalls Treibermodule nachladen. Sie empfängt die Aufrufe und gibt sie an das Treibermodul weiter, welches die konkrete Hardware anspricht (Abbildung 6.4).

Durch dieses Konzept muss der Anwendungsprogrammierer zur Compilezeit noch keine Entscheidung über den Netzwerkzugriff treffen. Erst der Benutzer konfiguriert das Programm zur Laufzeit und L.I.V.E. lädt die entsprechenden Treibermodule nach.

6.2 Scientific Visualization Toolkit

Die in Kapitel 4 entwickelte Anwendung soll sowohl durch haptisches Rendern als auch durch Virtual-Reality-Methoden den Benutzer beim Abbildungsprozeß unterstützen. Neben der Ansteuerung der haptischen Geräte durch L.I.V.E. wurde eine Bibliothek zur Realisierung einer VR-Anwendung gesucht.

Die zu realisierende Anwendung muss sowohl auf dem Virtual-Reality-System des Instituts als auch auf normalen Workstations und PCs ablauffähig sein. Da die verwendete VR-Umgebung des Zentralinstituts für Angewandte Mathematik von einem Unix-System der Firma SGI betrieben wird, sollen zumindest das hier eingesetzte Irix-Betriebssystem sowie die auf dem PC üblichen Betriebssysteme (Windows, Linux) unterstützt werden.

Die Software soll zwar VR-Systeme betreiben können, eine Festlegung auf das konkrete VR-System ist aber nicht sinnvoll. Es wäre wünschenswert, dass die Anwendung in verschiedenen VR-Systemen (zum Beispiel Holobench, CAVE, siehe 2.2.1) ablaufen könnte. Ausserdem müssen auch einfache Workstations oder ein PC in der Lage sein, die Anwendung auszuführen.

Da die Zusammenarbeit mit anderen Forschern wichtig erschien, sollte die gesuchte Lösung möglichst flexibel sein. Vor allem sollte die Bibliothek möglichst kostenfrei erhältlich sein. Anderenfalls wären Kooperationen mit anderen Gruppen problematisch gewesen.

Zu dem Zeitpunkt, als die Arbeit begonnen wurde, lag keine Software vor, die diese Bedingungen erfüllt hätte. Entweder war die Portabilität nur sehr eingeschränkt vorhanden (zum Beispiel bei Performer), es wurden keine VR-Systeme unterstützt (zum Beispiel von VTK) oder die Software war zu teuer (zum Beispiel WTK).

Aus diesem Grunde wurde eine eigene Softwarebibliothek mit dem Namen “Scientific Visualisation Toolkit” (kurz SVT) entwickelt, die im folgenden vorgestellt wird. SVT ist eine in C++ implementierte Klassenbibliothek, die sowohl Visualisierungs- als auch VR-Techniken realisiert.

6.2.1 Aufbau von SVT

Abbildung 6.5 zeigt einen Überblick über die verschiedenen Module von SVT.

Die Module können in zwei verschiedene Kategorien eingeteilt werden. Es existieren Kernmodule, die die grundlegende Funktionalität von SVT realisieren. Diese Module werden von allen Anwendungsprogrammen, die SVT nutzen, benötigt. Die Kernmodule sind: `svt_basics`, `svt_file_io`, `svt_system` und `svt_core`.

Darüber hinausgehende Funktionalitäten, wie zum Beispiel spezielle Visualisierungsmöglichkeiten für ein bestimmtes Anwendungsgebiet, werden in separate Module ausgelagert.

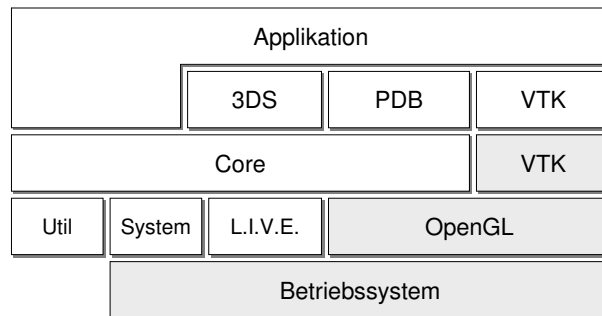


Abbildung 6.5: Übersicht über SVT

Diese Module werden jedoch nicht von allen Anwendungen benötigt. Beispiele für zusätzliche Spezialmodule sind `svt_pdb` oder `svt_vtk`.

Die Abspaltung der Spezialmodule kommt der Wartbarkeit sehr entgegen. Die Wartbarkeit wird durch die Nutzung von immer mehr externen Bibliotheken erschwert. Es steigt das Risiko, dass diese Bibliotheken für bestimmte Plattformen nicht zur Verfügung stehen und so das ganze Projekt nicht auf diese Plattformen portiert werden kann. In SVT sind in der Regel nur die Spezialmodule von externen Bibliotheken abhängig. Sollte eine dieser Bibliotheken auf einer Plattform nicht zur Verfügung stehen, so kann das jeweilige Spezialmodul nicht portiert werden. Das restliche SVT-System ist davon aber nicht betroffen und alle Anwendungen, die nicht von dem Spezialmodul abhängen, können problemlos auf der neuen Architektur ablaufen.

Ein weiterer Vorteil ergibt sich durch die minimierte Größe der Applikationen. Jede Anwendung nutzt nur genau die Module, die sie wirklich benötigt. Dies ist nicht nur wegen dem geringeren Platzbedarf und dem schnelleren Startverhalten der Applikation wünschenswert, auch die Distribution der Software über das Internet wird so deutlich erleichtert.

Im folgenden wird die Funktionalität der Kernmodule und ein kurzes SVT-Beispielprogramm erläutert. Später wird das für die vorliegende Arbeit ebenfalls eingesetzte Spezialmodul `svt_pdb` beschrieben.

svt_basics

`svt_basics` ist ein Kernmodul, in dem häufig benötigte grundlegende Klassen gesammelt werden. Dies sind zum Beispiel mathematische Klassen (Vektor- und Matrixklassen mit den dazugehörigen Manipulationsfunktionen) oder auch allgemeine Dinge wie Klassen zur Verarbeitung von Zeichenketten oder dynamische Arrays.

Weiterhin werden hier eigene SVT-Datentypen definiert. Dies ist notwendig, da in C und C++ die genaue Bitlänge der Datentypen nicht definiert ist. Eine Integer-Variable kann also 16, 32 oder gar 64 Bit breit sein. Dies hat jedoch zur Folge, dass der in ihr abspeicherbare

Wertebereich unterschiedlich groß ist. Ausserdem ist der Datenaustausch problematisch, sei es beim Abspeichern des Variableninhalts in einer Binärdatei oder beim Versand über das Netzwerk. SVT definiert daher eigene Datentypen, die nicht nur den Typ des Inhalts sondern auch die Bitlänge festlegen.

svt_system

In `svt_system` werden alle hardwareabhängigen Teile der Bibliothek gesammelt, so dass bei einer Portierung auf eine neue Rechnerarchitektur oder ein neues Betriebssystem nur dieses Modul angepasst werden muss.

Es hat sich herausgestellt, dass folgende Bereiche besonders plattformabhängig und daher von `svt_system` zu realisieren sind:

- **Benutzeroberfläche**

SVT muss Fenster für die Graphikausgabe auf der graphischen Benutzeroberfläche öffnen können. Dies ist genauso wie die Abfrage der Tastatur und der Maus plattformabhängig.

- **Threads**

Bei den Unix-Betriebssystemen hat sich in den letzten Jahren die Posix-Thread-Bibliothek als allgemeiner Standard durchgesetzt. Windows benutzt jedoch eine eigene, zu Posix inkompatible Lösung. Zum Thread-Management gehören auch Klassen wie `svt_semaphore`, die Threads blockieren und so den gleichzeitigen Zugriff auf bestimmte Ressourcen verhindern kann.

- **Zeitmessung**

Wichtig ist eine möglichst fein aufgelöste Zeitmessung im Millisekundenbereich, die ebenfalls nur plattformabhängig realisiert werden kann.

- **OpenGL-Erweiterungen**

OpenGL bietet ein flexibles Konzept zur Erweiterung des Funktionsumfanges an. Die OpenGL-Erweiterungen sind nachladbare Module, die den Funktionsumfang um optionale Komponenten ergänzen. Leider ist das Nachladen der Module nicht plattformunabhängig gelöst worden.

svt_file_io

In `svt_file_io` werden die Datei-Ein- und -Ausgaben gekapselt. Hierbei handelt es sich vor allem um die Klassen zum Einlesen und Ablegen von Graphikdateien wie zum Beispiel TGA oder BMP. Es sind hier aber auch die Klassen zum Einlesen von Konfigurationsdateien realisiert worden.

6.2.2 svt_core

svt_core ist das größte Modul im SVT System. Es nutzt die oben beschriebenen Module, um die Ansteuerung von VR-Systemen sowie den Szenengraphen zu realisieren.

Ansteuerung von VR-Systemen

Ein VR-System unterscheidet sich in vielen Komponenten von einer Graphikworkstation. Neben den eventuell vorhandenen haptischen Geräten liegen die Unterschiede im komplexeren Ausgabesystem für visuelle Informationen. Ein VR-System bestimmt die Betrachterposition und passt die graphische Darstellung an diese Position an. Im Gegensatz zu einer konventionellen Workstation kann dadurch erreicht werden, dass der Betrachter virtuelle Objekte an einer realen Position und mit realen Ausmassen wahrnimmt.

Um dies zu erreichen, muss der virtuelle Raum auf den realen Raum abgebildet werden. Hierzu wird ein realer Punkt als Ursprung des Weltkoordinatensystems ausgewählt, der für die Bestimmung von allen anderen Koordinaten der Bezugspunkt ist. Sowohl die Betrachterposition als auch die Objektpositionen werden relativ zu diesem Punkt angegeben.

Die Betrachterposition wird von einem Headtracking-System gemessen, welches die Position zunächst relativ zu der Lage seiner Basisstation angibt. Die Position dieses Gerätes muss relativ zum Weltkoordinatenursprung ausgemessen und so eine Matrix ${}_{World}M_{HT}$ aufgestellt werden. Mit deren Hilfe kann das Betrachterkoordinatensystem in das Weltkoordinatensystem abbildet werden: ${}_{World}M_{Viewer} = {}_{World}M_{HT} {}_{HT}M_{Viewer}$.

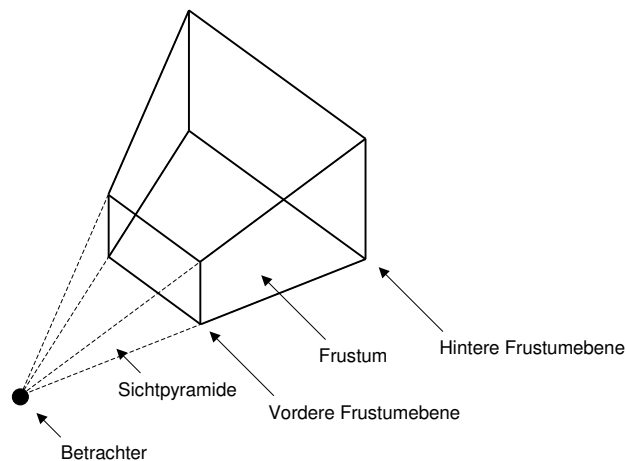


Abbildung 6.6: Sichtpyramide und Frustum

Dargestellt werden die virtuellen Objekte mit Hilfe von zweidimensionalen Bildschirmen durch eine perspektivische Projektion. Diese wird mit Hilfe einer Sichtpyramide durchgeführt (siehe Abbildung 6.6), wobei nur Objekte gezeichnet werden, die innerhalb der Pyramide liegen. Da die Sichtpyramide unendlich groß ist, wird diese verkleinert, indem zwei

zusätzliche Ebenen eingefügt werden. Eine vordere Ebene unterdrückt das Zeichnen von Objekten, deren Entfernung zum Betrachter zu gering ist, wohingegen eine hintere Ebene die zu weit entfernt liegenden Objekte unterdrückt. Der entstehende Raum wird Frustum genannt. Um das Frustum aufstellen zu können, muss sowohl die Position der Projektionsflächen (im Weltkoordinatensystem) als auch die Betrachterposition bekannt sein, wobei diese in einem VR-System ständig angepasst werden muss.

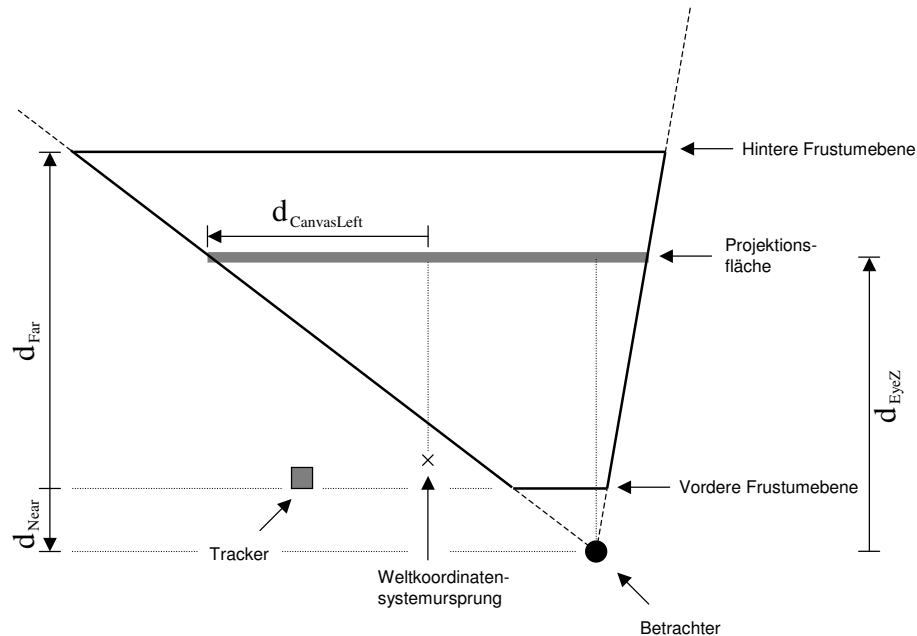


Abbildung 6.7: Projektion am Beispiel der Holobench

Die konkrete Situation am Beispiel der Holobench (das in dieser Arbeit eingesetzte VR-System) wird in Abbildung 6.7 dargestellt (zur Vereinfachung nur die Sicht von oben auf das System, eingezeichnet ist die senkrechte Projektionsfläche). Zur Konstruktion des Frustums wird zunächst die Sichtpyramide betrachtet. Sie entsteht aus den gedachten Geraden durch die Betrachterposition und die Eckkoordinaten der Projektionsfläche. Um diese Geraden bestimmen zu können, muss die reale Projektionsfläche relativ zum festgelegten Ursprung des Weltkoordinatensystems vermessen werden. Im vorliegenden System wurde die Position des Mittelpunktes und die Breite und Höhe der Displayflächen bestimmt. Hieraus ergibt sich die eingezeichnete Variable $d_{CanvasLeft}$, die der Position der linken Kante der Projektionsfläche relativ zum Mittelpunkt entspricht.

Aus der Position der Projektionsfläche im Weltkoordinatensystem und der Betrachterposition ergibt sich die Variable d_{EyeZ} , die den Abstand des Betrachters von der Fläche angibt. Mit ihrer Hilfe und mit Hilfe der frei zu wählenden Variablen für die zusätzlichen Frustumebenen (im Bild d_{Near} und d_{Far}) kann das Frustum aufgestellt werden.

Für die perspektivische Projektion wird jedoch nicht nur das Frustum benötigt, sondern ebenfalls eine Matrix, welche von den Objektkoordinaten in Betrachterkoordina-

ten wandelt (die sogenannte Modelview-Matrix). Durch das exakte Ausmessen der Projektionsflächen in Weltkoordinaten können Matrizen $_{World}M_{Proj}$ angegeben werden. Das Headtracking-System liefert eine Matrix $_{Viewer}M_{World} = _{World}M_{Viewer}^{-1}$, so dass sich die Modelview-Matrix für eine Projektionsfläche wie folgt ergibt:

$$M_{Modelview} = _{Viewer}M_{World} _{World}M_{Proj}$$

Die gerade genannten Matrizen $_{World}M_{Proj}$ und Positionsangaben sind variabel, so dass sich SVT auch an andere VR-Systeme anpassen lässt. Über Konfigurationsdateien kann die Anzahl der Projektionsflächen und ihre jeweilige Matrix und physikalische Größe angegeben werden.

Szenengraph

Ein Szenengraph ist eine hierarchische Beschreibung einer dreidimensionalen Szene (Abbildung 6.8). Komplexe Objekte werden mit Hilfe von Kindknoten in einem Graphen beschrieben. Komplexere Objekte können so aus einfacheren Objekten zusammengesetzt werden.

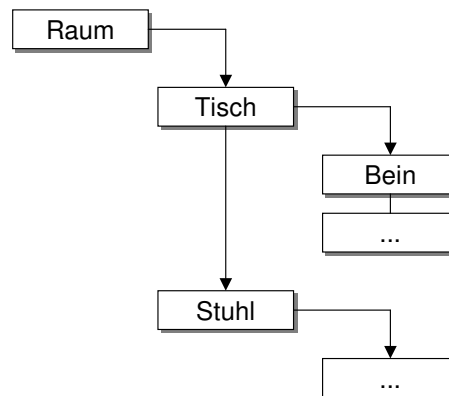


Abbildung 6.8: Hierarchische Beschreibung einer Szene

Ein Vorteil der hierarchischen Beschreibung liegt in diversen Vereinfachungen, die sich beim Umgang mit der Szene ergeben. Soll ein Objekt verschoben werden, so müssen nicht alle Kindobjekte ebenfalls verschoben und ihre Koordinaten angepaßt werden. Sie werden automatisch verschoben, da die Kindobjekte im lokalen Koordinatensystem des übergeordneten Objektes definiert sind.

Auch andere Verfahren profitieren von der hierarchischen Beschreibung. Soll eine Kollisionserkennung implementiert werden, so kann die hierarchische Struktur der Szene ausgenutzt werden, um die Anzahl an Kollisionstests möglichst klein zu halten. War der Kollisionstest für ein Objekt negativ, so müssen die an ihm hängenden Kindobjekte auch nicht

mehr betrachtet werden. Analog kann die hierarchische Beschreibung auch für Sichtbarkeitstests eingesetzt werden.

SVT implementiert daher einen solchen hierarchischen Szenengraph um die in einer Applikation enthaltenen graphischen Objekte zu verwalten.

In SVT basiert jede Szene auf Knotenobjekten des Szenengraphen. Die Basisklasse heisst `svt_node` und stellt ein abstraktes dreidimensionales Grundobjekt dar. Ein solches Grundobjekt besitzt diverse Eigenschaften, wie zum Beispiel eine Position (`svt_matrix`) oder diverse Materialeigenschaften wie Farbe (`svt_color`) oder Textur (`svt_texture`).

Jedes `svt_node`-Objekt kann sowohl mit einem Objekt der gleichen Hierarchiestufe als auch mit einem Kindobjekt verknüpft sein. Zum Zeichnen der gesamten Szene beginnt SVT beim Wurzelobjekt und durchläuft dann mit einer “depth-first”-Strategie rekursiv den Szenengraphen. Zuerst werden alle Kindknoten und dann alle Objekte gleicher Hierarchiestufe gezeichnet.

Das Zeichnen eines Objektes geschieht über eine Ausgaberoutine in `svt_node`, welche in den konkreten Klassen, die von `svt_node` abgeleitet sind, verschieden implementiert ist. Es stehen diverse Grundobjekte wie Dreiecke, Kugeln, Dreiecksgitter usw. als von `svt_node` abgeleitete Klassen zur Verfügung.

Volumen-Rendering

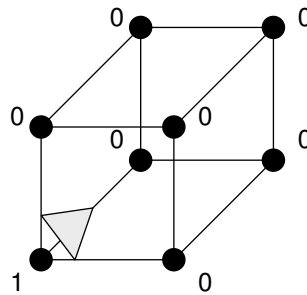
Da Volumen-Rendering in der vorliegenden Arbeit durch die Verarbeitung und Visualisierung von Dichteverteilungen ein Schwerpunkt ist, wurden in SVT verschiedene Techniken zum Volumen-Rendering implementiert.

Marching Cube Eine beliebte Methode ist die Wandlung der Volumendaten in Oberflächendaten, so dass sie leichter mit der heute zur Verfügung stehenden Graphikhardware visualisiert werden können. Sehr einfach können “Iso-Oberflächen” gebildet werden. Diese verbinden alle Punkte in einem Volumendatensatz, an denen ein gewisser Schwellwert erreicht wird.

Um Iso-Oberflächen zu erstellen, existieren verschiedene Algorithmen, von denen viele auf den sogenannten Marching-Cube-Algorithmus von Lorensen und Cline [35] zurückgehen. Dieser wurde auch in SVT implementiert.

Der Algorithmus legt über das Volumen ein dreidimensionales Gitter, so dass die Daten in kleine Würfel eingeteilt werden. An jedem Eckpunkt eines Würfels liegt genau ein Datenpunkt des Volumens. Wenn mindestens einer der Eckpunkte über und gleichzeitig mindestens einer unter dem Schwellwert liegt, dann muss die Oberfläche durch den Würfel hindurch verlaufen. Haben umgekehrt alle Eckpunkte geringere oder höhere Datenwerte, dann verläuft die Oberfläche nicht durch diesen Würfel.

Da jeder Würfel 8 Eckpunkte hat, gibt es $2^8 = 256$ mögliche Kombinationen, in denen die verschiedenen Punkte mal über oder unter dem Schwellwert liegen. Zwei dieser Kombinationsmöglichkeiten sind die beiden trivialen, in denen alle Eckpunkte über oder unter dem Wert liegen und somit der Würfel nicht zu der Oberfläche beiträgt.



(1,0,0,0,0,0,0,0)

Abbildung 6.9: Ein Würfel des Marching-Cube-Algorithmus

Lorensen und Cline haben die restlichen Kombinationen genau untersucht und festgestellt, dass maximal 4 Dreiecke entstehen können. Um diese Dreiecke aufstellen zu können, muss bekannt sein, welche der Kanten des Würfels von der Oberfläche geschnitten wird. Dies hängt von den Eckpunkten und ihrem Datenwert verglichen mit dem Schwellwert ab. Es existieren zwei Fälle: der Datenwert ist kleiner oder gleich, dann wird dem Eckpunkt eine 0 zugeordnet, andernfalls eine 1. Da acht Eckpunkte existieren, kann die gesamte Situation eines Würfels in einem 8-Bit-Wert abgespeichert werden (Abbildung 6.9).

Mit Hilfe einer Tabelle können die Dreiecke sehr effizient bestimmt werden. In dieser werden für jeden der 256 Fälle die Kanten angegeben, die die Oberfläche schneiden. Die exakte Position der Dreieckspunkte auf den Kanten ergibt sich durch lineare Interpolation zwischen den Volumenwerten auf den zugehörigen Eckpunkten des Würfels.

Der Algorithmus hat den Nachteil, dass er bei großen Volumendaten sehr große Dreiecksnetze mit vielen Elementen erstellt (Abbildung 6.10). An die Erstellung der Dreiecke knüpft er keine weiteren Bedingungen, außer dass das Volumen an dieser Stelle den oben genannten Schwellwert erreichen muss. Durch das reguläre Abschreiten des gesamten Volumens entstehen so schnell Netze mit Millionen von Dreiecken. Oft beschreiben diese Dreiecke jedoch größere ebene Flächen, bei denen viele Dreiecke gar keine visuelle Information mehr repräsentieren. Hier könnten Dreiecke zu größeren Dreiecken zusammengefasst und so die Gesamtmenge deutlich reduziert werden.

Erreicht werden kann dies, indem Algorithmen zur Vereinfachung von Dreiecksnetzen angewendet werden. Da das Problem der zu großen Dreiecksnetze nicht nur bei Iso-Oberflächen sondern auch bei anderen Problemen (zum Beispiel Terrainrendering) auftritt, existieren zu diesem Problem verschiedene Lösungsansätze (unter anderem die “Progressive Meshes” [24] von Hoppe).

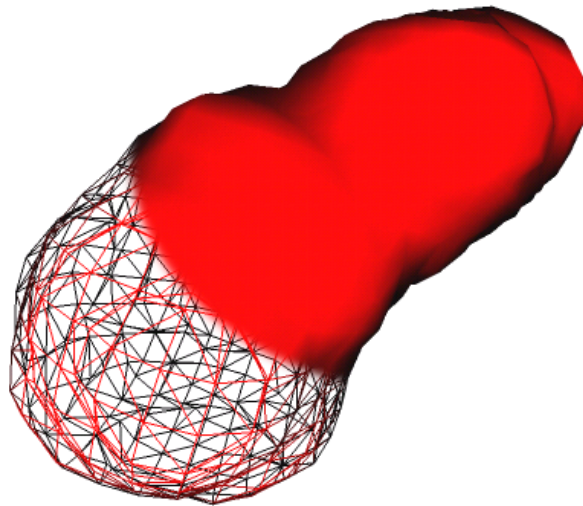


Abbildung 6.10: Oberflächendarstellung eines Volumens

Ein grundsätzlicheres Problem aller Oberflächendarstellungen stellt die Wahl eines geeigneten Schwellwertes dar. Die Wahl hängt davon ab, was der Betrachter mit der Iso-Oberfläche erkennen möchte, es gibt kein allgemeines Kriterium und auch keine Berechnungsformel für diesen Wert. Eine falsche Wahl des Schwellwertes kann es für den Benutzer unmöglich machen, gesuchte Strukturen zu erkennen, weil der Algorithmus Teile des Volumens nicht mehr darstellt.

Dreidimensionale Texturen In der Regel werden unter dem Begriff “Texturen” zweidimensionale Bilder verstanden, die auf dreidimensionale Polygonobjekte projiziert werden können. Um die Lage der Textur auf dem dreidimensionalen Polygon spezifizieren zu können, existieren zweidimensionale Texturkoordinaten, die jedem Punkt des Polygons zugeordnet werden können.

Dreidimensionale Texturen sind analog dazu dreidimensionale Volumendaten, die auch wieder auf Polygonobjekte projiziert werden. Die Texturkoordinaten müssen in diesem Fall dreidimensional sein. Sichtbar ist auf dem Polygon später nur ein Schnitt durch die Volumendaten. Eindeutig definiert ist der Schnitt durch die dreidimensionalen Texturkoordinaten an jedem Punkt des Polygons.

Sollen die dreidimensionalen Texturen für Volumen-Rendering genutzt werden, so kann die von Wilson et al. [63] vorgeschlagene Methode eingesetzt werden. Sie verwendet für fast alle Aufgaben die Graphikhardware und ist daher ausreichend schnell.

In einem ersten Schritt wird eine Textur aus dem Volumendatensatz erzeugt. Hierbei werden mit Hilfe einer geeigneten Transferfunktion aus den Volumendaten Farb- und Transparenzwerte berechnet. Zur Vereinfachung der folgenden Erläuterungen wird davon aus-

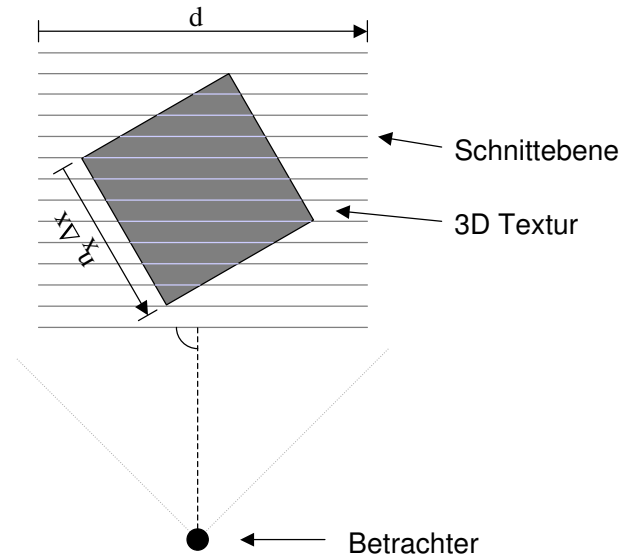


Abbildung 6.11: Volumerendering mit 3D-Texturen

gegangen, dass das darzustellende Volumen genau im Ursprung des Koordinatensystems liegt.

Hat der Volumendatensatz die Auflösung (n_x, n_y, n_z) und eine Voxelgröße $(\Delta x, \Delta y, \Delta z)$, so ergibt sich eine Gesamtgröße des darzustellenden Volumenwürfels von $(n_x \Delta x, n_y \Delta y, n_z \Delta z)$. Es wird ein umgebender Würfel konstruiert, der so groß gewählt wird, dass der Volumenwürfel mit jeder Drehung immer komplett hineinpasst. Der umgebende Würfel besitzt die Kantenlänge d , welche sich aus der Länge der Diagonale durch den Volumenwürfel bestimmen lässt — $d = \sqrt{n_x \Delta x * n_x \Delta x + n_y \Delta y * n_y \Delta y + n_z \Delta z * n_z \Delta z}$.

Wurde in einem ersten Schritt die Textur mit Hilfe der Transferfunktionen aus dem Volumendatensatz erzeugt, so wird danach eine Reihe von Quadraten senkrecht zur Strecke vom Mittelpunkt zum Betrachter aufgestellt (siehe Abbildung 6.11). Jedes dieser Quadrate repräsentiert einen Schnitt durch den umgebenden Würfel mit einer Seitenlänge d .

Die dreidimensionalen Texturen werden auf die Quadrate projiziert. Zunächst werden dazu die Texturkoordinaten in der ursprünglichen Ausrichtung aufgestellt. Ist der Mittelpunkt des Volumenwürfels im Ursprung des Koordinatensystems, so muss dem Punkt $(-n_x \Delta x / 2, -n_y \Delta y / 2, -n_z \Delta z / 2)$ die Texturkoordinate $(0, 0, 0)$ zugewiesen werden. Analog besitzt der Punkt $(n_x \Delta x / 2, -n_y \Delta y / 2, -n_z \Delta z / 2)$ die Texturkoordinate $(1, 0, 0)$.

Die Texturkoordinaten werden aber an den Eckpunkten der Quadrate angegeben. Dazu müssen die Koordinaten noch geeignet interpoliert werden. Die Eckpunkte der Quadrate sind gleichzeitig die Eckpunkte des umgebenden Würfels, besitzen also die Koordinaten $(\pm d/2, \pm d/2, \pm d/2)$. Die Texturkoordinaten werden mit (s, t, r) bezeichnet. Da $s(0) = \frac{1}{2}$

und $s(n_x \Delta x / 2) = 1$, gilt für die linke Seite:

$$s(-d/2) = \frac{1}{2} \left(1 - \frac{d}{n_x \Delta x} \right)$$

analog die rechte Seite:

$$s(d/2) = \frac{1}{2} \left(1 + \frac{d}{n_x \Delta x} \right)$$

Die t und r Koordinaten können analog aufgestellt werden. Die r Koordinate muss für jedes Quadrat angepasst werden, da diese ja in der z Achse verschoben aufgestellt worden sind.

Wurden die Eckpunkte der Quadrate mit geeigneten Texturkoordinaten versehen, dann kann der Volumendatensatz dargestellt werden. Soll der Datensatz gedreht werden können, so müssen die Texturkoordinaten noch mit einer Rotationsmatrix multipliziert werden. Dies kann durch Ausnutzen der Graphikhardware vereinfacht werden. Die Graphikhardware kennt einen Texturmatrix-Stack, der für die Berechnung der Texturkoordinaten benutzt wird. Auf diesem kann die Rotationsmatrix abgelegt werden, die Texturkoordinaten werden dann von der Hardware mit dieser Matrix multipliziert. Es darf auch bei einer Rotation die Lage der Quadrate nicht verändert werden — sie müssen immer parallel zur Projektionsebene ausgerichtet sein.

Nach der Implementierung der beschriebenen Methode störten zunächst die auftretenden Artefakte. Es war jede Schnittebene klar zu erkennen. Diese Artefakte konnten mit Hilfe von verschiedenen “Blending”-Techniken verringert werden, wobei diese Filter darstellen, die die einzelnen Schnitte ineinander überblenden.

Zweidimensionale Texturen Da einige Graphikkarten keine dreidimensionalen Texturen unterstützen, ist es sinnvoll, die gerade beschriebene Technik abzuwandeln und zweidimensionale Texturen zu verwenden.

Dazu müssen drei Stapel mit Quadraten angelegt werden. Ein Stapel wird analog zu dem gerade beschriebenen Verfahren aufgebaut, also in Richtung der Z -Achse. Ein zweiter Stapel schneidet das Volumen in Richtung der X - und ein dritter in Richtung der Y -Achse. Auf jedes der Quadrate wird eine Textur abgebildet, welche den passenden Schnitt durch das Volumen darstellt. Hierbei werden auch die oben beschriebenen Transferfunktionen eingesetzt.

Im Gegensatz zu den dreidimensionalen Texturen wird aber nicht die Textur gedreht, sondern die Stapel. Es ist immer nur der Stapel sichtbar, dessen Normale nahezu parallel zum Betrachtersehstrahl ausgerichtet ist. Es kann so nicht ganz verhindert werden, dass die Kanten der Stapel in bestimmten Fällen sichtbar werden. Ein weiterer Nachteil ist der hohe Speicherplatzverbrauch, da das Volumen dreimal im Texturspeicher abgelegt wird.

Vorteile liegen in der Geschwindigkeit. Die Darstellung ist auch auf langsameren Computern noch ausreichend schnell, da der Speicherzugriff auf zweidimensionale Texturen wesentlich schneller abläuft als auf dreidimensionale [62].

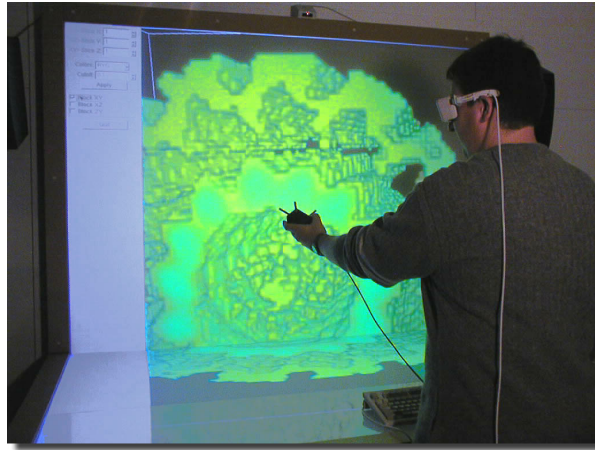


Abbildung 6.12: Volumerendering mit Volslice3D

Mit SVT und der Implementierung dieser Methode wurde das Programm Volslice3D [64] entwickelt. Es visualisiert Elektronenmikroskopie-Dichteverteilungen im Situs-Format (siehe Kapitel 5.2). Als Benutzeroberflächenbibliothek wurde GLUI[42] eingesetzt (siehe Abbildung 6.12). Das Programm kann nicht nur das gesamte Volumen darstellen, sondern auch einzelne Schnitt-/ebenen. Diese Schnittebenen können durch das Volumen verschoben werden und zeigen den gerade aktuellen Schnitt an.

Eingabegeräte und L.I.V.E.-Integration

SVT setzt zur Abfrage von Eingabegeräten die in Abschnitt 6.1 beschriebene Bibliothek L.I.V.E. ein. Sie stellt für jedes Eingabegerät Datenkanäle zur Verfügung, die jeweils einen Datenstrom eines bestimmten Typs liefern. Dieses Konzept wurde in SVT auf mehrere C++ Klassen abgebildet.

Es existiert eine Klasse `svt_device`, welche genau einem L.I.V.E.-Gerät entspricht. Jedes Objekt dieser Klasse besitzt eine Menge von `svt_channel`-Objekten, welche jeweils einen L.I.V.E.-Datenkanal kapseln. Wird ein `svt_device`-Objekt erzeugt, so wird dem Konstruktor der Name des konkreten Gerätes, das das Objekt repräsentieren soll, übergeben. SVT teilt L.I.V.E. daraufhin mit, dass es das Gerät mit dem angegebenen Namen benutzen möchte und erfragt die Anzahl an Datenkanälen, die das Gerät zur Verfügung stellt. Ist ein solches Gerät vorhanden, dann legt SVT an dieser Stelle die benötigte Anzahl an `svt_channel`-Objekten an.

Da L.I.V.E. für die Geräte keine feste Funktionszuordnung trifft, kann SVT die `svt_device`-Objekte frei übergeordneten Objekten zuweisen. Es existiert eine

svt_tool-Klasse, welche allgemein interaktive Manipulationen an SVT-Objekten organisiert. Abgeleitet davon wurde die svt_move_tool-Klasse implementiert, die speziell für Manipulationen der Position zuständig ist.

Jedes svt_move_tool benötigt ein svt_device-Objekt, um die Manipulation interaktiv durchführen zu können. Welches svt_device-Objekt hier festgelegt wird, hängt von der jeweiligen Anwendung ab. SVT sieht vor, dass der Gerätenamen per Konfigurationsdatei abgefragt werden kann. So kann der Anwender auch ohne Ändern des Quellcodes mit einer bestehenden Anwendung verschiedene Geräte einsetzen.

6.2.3 SVT Beispielprogramm

Es folgt ein kurzes Beispielprogramm. Das Programm zeichnet zwei Kugeln, von denen eine mit der Maus bewegt werden kann.

```
#include <svt_scene.h>
#include <svt_scenegraph.h>
#include <svt_move_tool.h>
#include <svt_init.h>

// Beispielszene
class example_scene : public svt_scene
{
    svt_scenegraph* pScenegraph;
    svt_sphere* pSphereA;
    svt_sphere* pSphereB;
    svt_move_tool* pMoveTool;

    buildScene()
    {
        pScenegraph = new svt_scenegraph();
        pSphereA = new svt_sphere();
        pScenegraph->add(pSphereA);
        pMoveTool = new svt_move_tool(pSphereA);
        pSphereB = new svt_sphere();
        pScenegraph->add(pSphereB);
        setScenegraph(pScenegraph);
    }
};

// Hauptroutine
int main(int argc, char* argv[])
{
    example_scene* pScene = new example_scene();
    svt_setScene(pScene);
    svt_init(argc, argv);
    return 0;
}
```

```
};
```

6.2.4 svt_pdb

Die Funktionalität des SVT-Zusatzmoduls `svt_pdb` konzentriert sich auf die Visualisierung von PDB-Dateien. Diese Dateien beschreiben ein komplettes Molekül, das aus einer Menge von Atomen zusammengesetzt ist. `svt_pdb` kann eine solche Datei einlesen, visualisieren, manipulieren und auch wieder auf der Festplatte ablegen.

Berechnung der Atombindungen

Wie schon in Kapitel 2.1.1 beschrieben wurde, sind die CONNECT-Einträge in den PDB-Dateien oft nicht vorhanden oder nicht korrekt. Daher müssen auf der Grundlage der Atomentfernungen die Bindungen geschätzt werden. Es wird von einer Bindung zwischen zwei Atomen ausgegangen, wenn der Abstand einen Schwellwert ϵ unterschreitet. Für ϵ wird in der Regel der maximale van der Waals Radius $2,0\text{\AA}$ multipliziert mit 1,2 eingesetzt.

Die Suche nach den Atombindungen ist nicht trivial. Sollen anhand des gerade beschriebenen Abstandskriteriums alle Bindungen berechnet werden, so entsteht durch die notwendigen Vergleiche aller Atome eine Komplexität von $O(N^2)$.

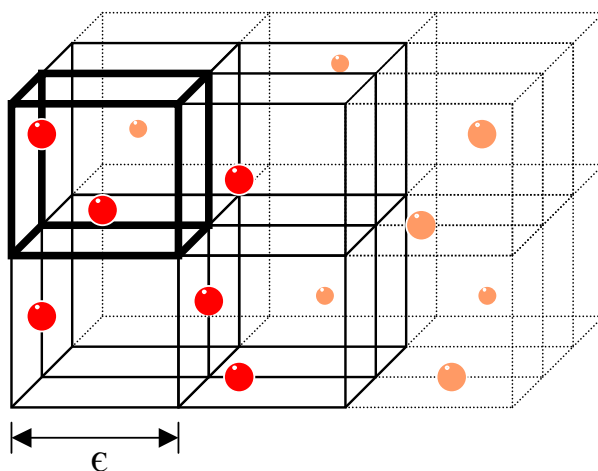


Abbildung 6.13: Space-Partitioning zur Berechnung der Atombindungen

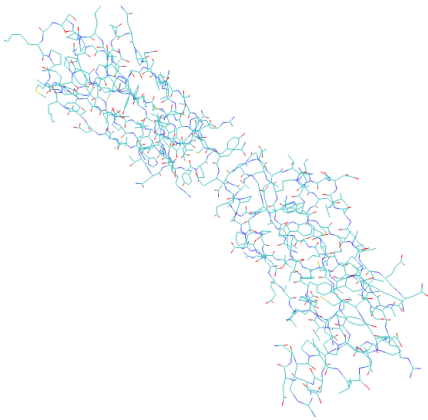
`svt_pdb` wendet als Optimierung eine sogenannte "Space-Partitioning"-Strategie an. Hier wird der gesamte Suchraum in kleine Würfel eingeteilt, in die die Atome einsortiert werden. Die Ausdehnung eines Würfels muss größer als der größte mögliche Abstand zweier gebundener Atome (also 2ϵ) sein. Auf diese Weise kann sichergestellt werden, dass die Atome eines Würfels nur Bindungen mit Atomen aus dem gleichen oder direkt benachbarten Würfeln eingehen können.

Werden alle Würfel systematisch untersucht, so müssen die Atome eines Würfels nur mit den Atomen von sieben benachbarten Würfeln verglichen werden (siehe Abbildung 6.13).

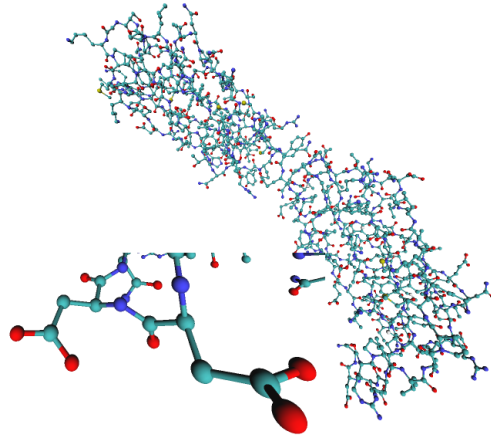
Zeichenmodi

Um die Daten einzulesen und darzustellen, steht die C++ Klasse `svt_pdb` zur Verfügung. Über diese werden alle Manipulationen sowohl an der Darstellung als auch an den Daten vorgenommen. Jedes Molekül besteht aus einer Menge von Atomen. Analog steht eine weitere Klasse `svt_atom` zur Verfügung, die jeweils ein einzelnes Atom verwaltet. `svt_atom` stellt über Memberfunktionen alle Informationen aus dem zugehörigen ATOM Record der PDB-Datei zur Verfügung. Dies sind zum Beispiel Atomtyp, Index, Position, Chain ID oder auch Residuumzugehörigkeit.

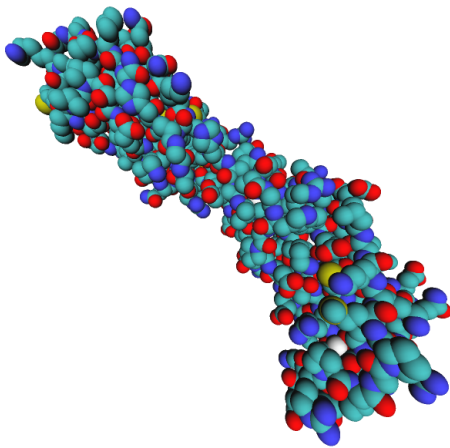
Über die Funktion `svt_pdb::setDisplayMode()` können folgende Darstellungen global für das ganze Molekül eingestellt werden:



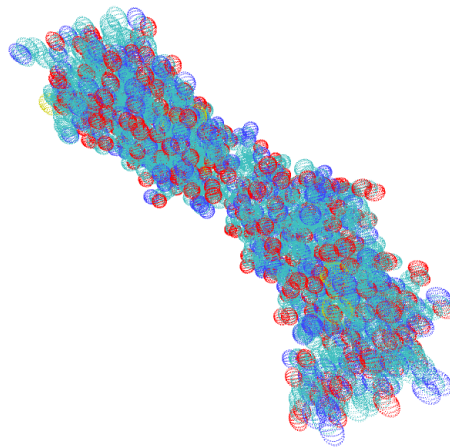
Liniendarstellung



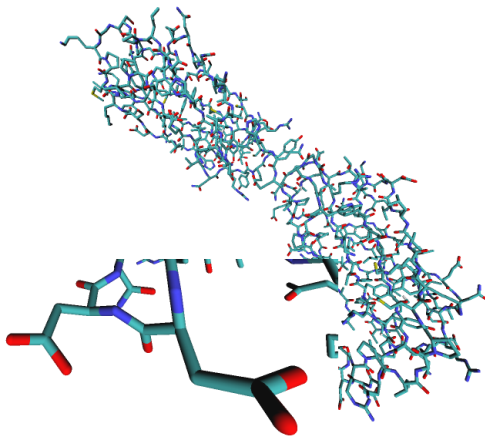
CPK



van der Waals Kugeln



"dotted" van der Waals Kugeln



Licorice



Tube

“Tube”-Darstellung

Der Tube Modus stellt mit Hilfe eines Splines eine “Röhre” durch alle sogenannten “Backbone”- beziehungsweise C_α -Atome des Moleküls dar. Wie schon in Kapitel 2.1.1 beschrieben, bietet sich dieser Modus vor allem für eine reduzierte Darstellung komplexer Moleküle an. Der hier implementierte Spline ist ein Catmull-Rom-Spline (kurz CR-Spline [17]). Er gehört zur Gruppe der approximierenden Splines.

Allgemein kann ein CR-Spline eine Kurve durch die Punkte P_1, \dots, P_{m-1} legen, wenn insgesamt P_0, \dots, P_m Punkte zur Verfügung stehen. CR-Splines bestehen aus Teilstücken, die immer den Bereich zwischen zwei Punkten P_{i-1} und P_{i-2} beschreiben, jedoch auf vier Punkten P_i, \dots, P_{i-3} definiert sind (mit $3 \leq i \leq m$). Die approximierten Zwischenpunkte werden durch den Parameter t angesprochen, der aus dem Intervall $[0, \dots, 1]$ entnommen wird. Jeder Zwischenpunkt $Q_i(t)$ eines Teilstücks i ist ein Vektor, für den gilt:

$$Q_i(t) = \underbrace{\begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix}}_T \underbrace{\frac{1}{2} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}}_{M_{CR}} \begin{bmatrix} P_{i-3} \\ P_{i-2} \\ P_{i-1} \\ P_i \end{bmatrix}$$

mit

$$B = T M_{CR} = [B_{-3} \ B_{-2} \ B_{-1} \ B_0]$$

also

$$\begin{aligned} Q_i(t) &= [B_{-3} \ B_{-2} \ B_{-1} \ B_0] \begin{bmatrix} P_{i-3} \\ P_{i-2} \\ P_{i-1} \\ P_i \end{bmatrix} \\ &= B_{-3}P_{i-3} + B_{-2}P_{i-2} + B_{-1}P_{i-1} + B_0P_i \end{aligned}$$

Über ein Austauschen der Matrix M_{CR} können auch andere Splintypen eingesetzt werden. Weiterhin kann durch eine Verdopplung des ersten und letzten Punktes erreicht werden, dass die approximierende Kurve doch durch alle P_0, \dots, P_m (und nicht nur durch P_1, \dots, P_{m-1}) Punkte führt.

In Abbildung 6.14 wird ein Molekül (alanin.pdb) graphisch dargestellt. Alle Atome wurden zunächst mit Hilfe von Linien gezeichnet. Danach wurde der gerade vorgestellte Tube-Spline eingezeichnet, sowie die C_α -Atome durch größer skalierte VDW-Kugeln hervorgehoben.

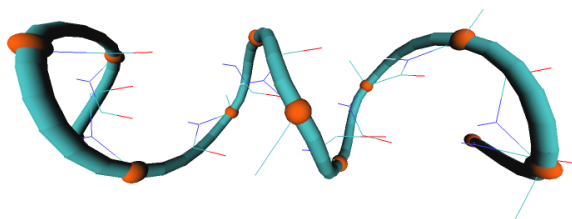


Abbildung 6.14: Tube-Modus

Kombination der Darstellungsmodi

In vorangegangenen Abschnitten wurden die verschiedenen Darstellungsmodi immer global auf das ganze Molekül angewendet. Es können aber auch den einzelnen Atomen unterschiedliche Darstellungsmodi zugewiesen werden. Dazu bietet `svt_pdb` verschiedene Methoden an, mit denen gezielt auf die Atome des Moleküls zugegriffen werden kann.

Aufzählungsfunktionen durchsuchen das Molekül hierzu nach Atomen, die ein bestimmtes Kriterium erfüllen. Es wird so lange jeweils ein Atom zurückgegeben, so lange noch Atome das geforderte Kriterium erfüllen.

Trivial ist die einfachste Aufzählungsfunktion, welche die Atome mit aufsteigender Indexnummer zurückliefert. Es existieren des weiteren Funktionen, die nach allen Atomen eines bestimmten Typs, eines bestimmten Residuums (Name oder Sequenznummer), einer Kette (Chain) oder einer Segment ID suchen.

Um Darstellungsmodi mischen zu können, kann bei den gefundenen Atomen der gewünschte Darstellungsmodus eingestellt werden. Das folgende Beispiel-Listing stellt für alle Atome eines Moleküls zunächst den Linienmodus ein, verändert danach aber für alle Kohlenstoffatome den Darstellungsmodus in VDW:

```
svt_pdb* pPDB = new svt_pdb("alanin.pdb");
svt_atom* pAtom = NULL;

// alle Atome in Liniendarstellung
pPDB->setDisplayMode(SVT_MODE_LINE);

// alle "C" Atome in VDW Modus schalten
pPDB->resetAtomEnum();
while(pAtom= pPDB->enumAtomType("C") != NULL)
{
    pAtom->setDisplayMode(SVT_MODE_VDW);
}
```

6.3 Testanwendung

Vor der Implementierung des eigentlichen Anwendungsprogramms wurde eine kleinere Testanwendung erstellt. Ziel war es, die Teilbibliotheken ausreichend zu erproben, um eventuell noch vor der Erstellung der Hauptanwendung Designänderungen am Klassen-aufbau vornehmen zu können.

Hierzu wurde im Rahmen der Ausstellung Physik 2000 ein Demonstrationsprogramm entwickelt, das den Besuchern der Ausstellung die Funktionsweise eines Rastertunnelmikroskops (kurz RTM) näherbringen sollte. Es sollte das direkte Manipulieren von Atomen mit der Mikroskopspitze des RTM in den Mittelpunkt gestellt werden.

Das Programm zeigt virtuell eine solche Mikroskopspitze und erlaubt es dem Benutzer, sie umherzubewegen. Auf einer ebenen Fläche werden einige Atome aufgebracht, die der Besucher mit der Spitze bewegen kann (Abbildung 6.15).

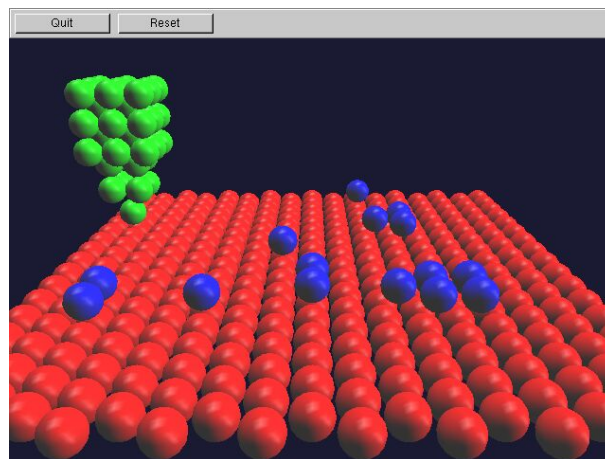


Abbildung 6.15: Testanwendung

Da das Programm mit Hilfe der oben vorgestellten Bibliotheken SVT und L.I.V.E. entwickelt wurde, stand es nicht nur für die Ausstellung zur Verfügung, sondern konnte danach auch auf dem am Zentralinstitut für Angewandte Mathematik vorhandenen VR-System vorgeführt werden.

Um die Demonstration für den Besucher besonders leicht verständlich zu machen, sollte die Anwendung in der Lage sein, die an der simulierten Mikroskopspitze auftretenden Kräfte mit Hilfe von haptischen Geräten auszugeben. Hierzu wurde die Bibliothek L.I.V.E. benötigt und die für den Einsatz am VR-System notwendige Netzwerkansteuerung implementiert.

Die Kraftberechnung für das haptische Rendern wurde mit Hilfe des Lennard-Jones-Potentials [3] durchgeführt:

$$V_{LJ}(d) = 4\epsilon \left[\left(\frac{\sigma}{d} \right)^{12} - \left(\frac{\sigma}{d} \right)^6 \right]$$

wobei d den Abstand zwischen den zwei Atomen angibt (ϵ und σ sind konstante Parameter). Mit Hilfe dieses Potentials können die Kräfte zwischen zwei Atomen berechnet werden.

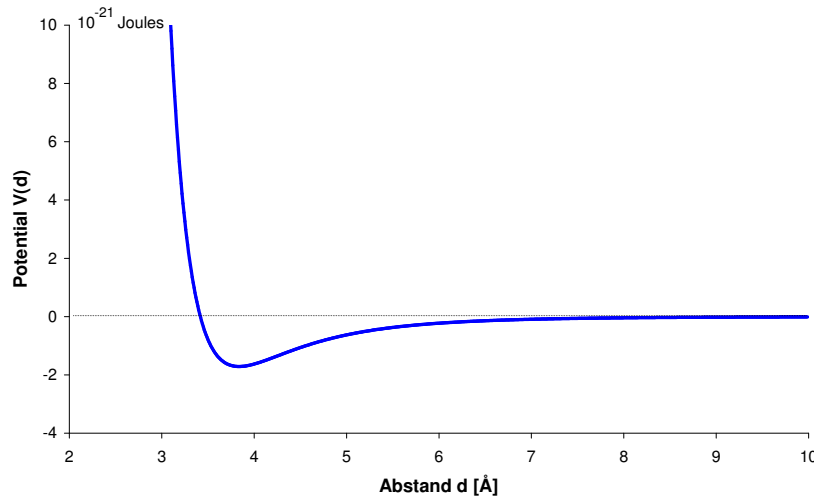


Abbildung 6.16: Lennard-Jones-Potential (Argon)

In Abbildung 6.16 ist der Funktionsgraph des Lennard-Jones-Potentials für Argon dargestellt (Parameter $\epsilon = 1,7119 \cdot 10^{-21}$ Joules und $\sigma = 3,418$ Ångström, mit 1 Ångström = 10^{-10} Meter). Werden zwei Atome aus großer Entfernung aufeinander zu bewegt, so entsteht zunächst ein anziehendes Potential. Das Potential sinkt immer weiter ab, bis es bei einem Abstand von zirka 3,85 Ångström Entfernung sein Minimum erreicht. Wird der Abstand noch weiter verringert, so steigt das Potential wieder stark an, so dass schnell eine abstossende Kraft entsteht. Atome können ihren Abstand also nicht auf 0 verringern.

Es wird davon ausgegangen, daß die Newtonschen Bewegungsgleichung auch für Atome weiterhin gilt, also:

$$\mathbf{F} = m \mathbf{a}$$

Werden nur Teilchen gleichen Typs betrachtet, so wird üblicherweise die Masse gleich Eins gesetzt, so daß sich für diesen Fall $\mathbf{F} = \mathbf{a}$ ergibt. Über die Ableitung des Lennard-Jones-Potentials kann die Kraft \mathbf{F} und somit auch die Beschleunigung \mathbf{a} der Teilchen berechnet werden.

Mit Hilfe des Verlet-Algorithmus [58] und der Beschleunigung, sowie der aktuellen Atompositionen, können die nächsten Atompositionen berechnet werden:

$$\mathbf{r}(t + \delta t) = 2\mathbf{r}(t) - \mathbf{r}(t - \delta t) + \delta t^2 \mathbf{a}(t)$$

Hierbei sei $\mathbf{r}(t)$ die Position eines Teilchens zum Zeitpunkt t . $\mathbf{a}(t)$ sei die Beschleunigung zum Zeitpunkt t dieses Teilchens. Neben der aktuellen Position geht auch die vorherige Position des Teilchens $\mathbf{r}(t - \delta t)$ in die Berechnung ein. Wichtig ist hierbei, dass die Schätzung der nächsten Position nur dann realistisch ausfällt, wenn der Parameter δt nicht zu groß ist (Verlet gibt hier $\delta t \approx 10^{-14} \text{ s}$ an).

Die durchgeführten Berechnungen haben nicht den Anspruch, eine ernstzunehmende Simulation der interatomaren Zusammenhänge bei dem Einsatz eines RTMs zu sein, sondern dienen nur Demonstrationszwecken.

Die Kräfte müssen zwischen allen Atomen berechnet werden, was zu einer Komplexität von $O(N^2)$ führt. Um hohe Kraftberechnungsraten zu erreichen, wurde vor der eigentlichen Kraftberechnung die Entfernung zwischen den Atomen untersucht. Nur wenn der Abstand unterhalb eines Schwellwertes lag (hier 8 Ångström), wurde die Berechnung fortgesetzt. Das Lennard-Jones-Potential nimmt bei größeren Abständen immer weiter ab, so daß bei einer größeren Entfernung das Potential vernachlässigt werden kann.

Es sind nur die auf der Fläche aufgebrachten Atome beweglich. Es werden daher nur für diese und für die Mikroskopspitze die Kräfte berechnet. In diese Berechnungen wurden auch die Atome der Ebene einbezogen, so dass die frei beweglichen Atome durch die Anziehungskraft festgehalten werden.

Durch diese Optimierungen war es möglich, auch auf einem PC hohe Kraft-Update-Raten von über 500 Hertz zu erreichen. Der PC kontrolliert das Kraftrückkopplungsgerät und zeichnet ebenfalls die Graphik. So konnte ein transportables Demonstrationssystem realisiert werden, das nur aus einem einzelnen PC und dem haptischen Gerät besteht.

Im Verlauf der Entwicklung dieser Testanwendung konnten noch Schwachstellen an den Bibliotheken SVT und L.I.V.E. behoben werden, so dass im Anschluss daran die nun vorzustellende Anwendung SenSitus entwickelt werden konnte.

6.4 SenSitus

Nachdem die Teilkomponenten beschrieben worden sind, soll nun das mit ihrer Hilfe entwickelte Anwendungsprogramm SenSitus vorgestellt werden.

6.4.1 Implementierung

SenSitus nutzt die oben beschriebene Klassenbibliothek SVT zur Visualisierung der hier auftretenden Datensätze. Dies sind zum einen die hochaufgelösten Molekülstrukturdaten in Form von PDB-Dateien. Diese werden mit Hilfe des Zusatzmoduls `svt_pdb` visualisiert. Des weiteren werden die dreidimensionalen Dichteverteilungen mit den Volumen-

Rendering-Klassen aus dem `svt_core`-Modul dargestellt. Die Ansteuerung der VR-Geräte und des Krafterückkopplungsgerätes Phantom übernimmt die Bibliothek L.I.V.E. Die Benutzeroberfläche wird mit Hilfe der plattformunabhängigen Bibliothek QT realisiert (Abbildung 6.17).

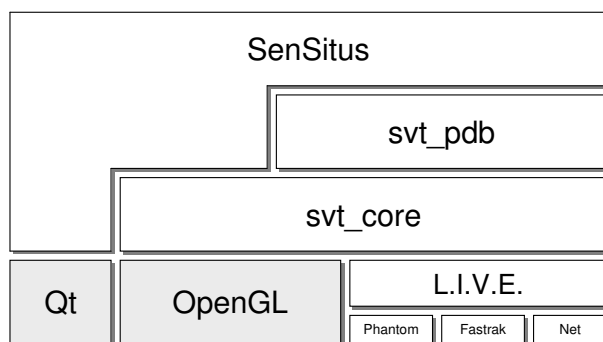


Abbildung 6.17: Aufbau von SenSitus

Da in der Anwendung verschiedene Dokumente zu verarbeiten sind, wurde die Klassenstruktur auf eine zentrale Dokumenten-Basisklasse und ihre Ableitungen ausgerichtet. Allgemeine Funktionen wie Positionsänderungen oder Änderungen des Darstellungszustandes konnten in der Basisklasse implementiert werden. Von dieser allgemeinen Klasse wurden spezielle Implementierungen für die Molekülstrukturdaten und die Dichteverteilungen abgeleitet. Ebenso wurden zusätzliche Dokumente wie eine Schnittebenenklasse entwickelt.

Diese spezielleren Klassen müssen diverse Memberfunktionen implementieren. Unter anderem werden Funktionen zum Laden und Speichern des Dokumentes oder zum Anzeigen eines Dialoges zur Konfiguration der Darstellung realisiert. Darüber hinaus können die Dokumente auch die graphische Benutzeroberfläche um eigene Menüs erweitern.

Neben den Dokumentenklassen existieren spezielle Klassen zur Kraftberechnung. Wird von der SVT-Klasse `svt_adv_force_algo` abgeleitet, so kann eine eigene Kraftberechnung für das haptische Rendern implementiert werden. In SenSitus wurde eine Klasse abgeleitet, die die in Kapitel 4 entwickelte Kraftberechnungsvorschrift umsetzt. Die zuständige Memberfunktion der Klasse `svt_adv_force_algo` läuft parallel zum Hauptprogramm in einem separaten Thread, so dass die Kraftberechnung von zusätzlichen Prozessoren in einem Shared-Memory-Parallelrechner automatisch profitiert.

Eine SVT-Klasse kapselt die Interaktion mit dem SVT-System und bietet grundlegende Funktionen zur Manipulation des Szenengraphen oder auch für das Anfertigen von Screenshots an.

6.4.2 Benutzerführung

SenSitus öffnet beim Start zwei Fenster. Das erste Fenster beinhaltet die graphische Darstellung und wird von SVT erzeugt, das andere Fenster zeigt die Benutzeroberfläche des Programms (Abbildung 6.18).

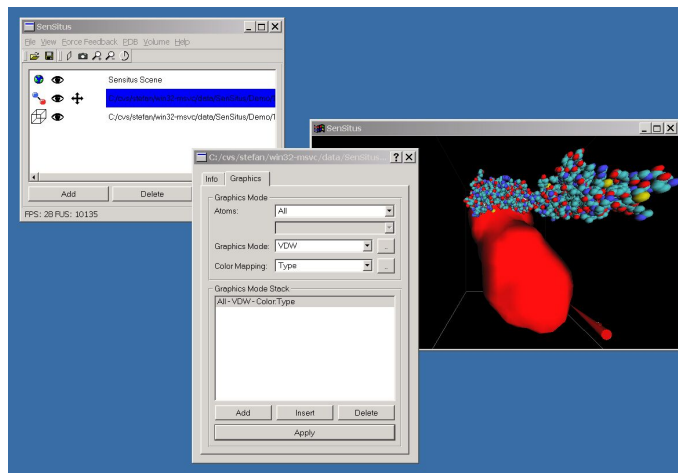


Abbildung 6.18: SenSitus

Hauptfenster

Die Benutzeroberfläche von SenSitus, dargestellt in Abbildung 6.19, zeigt eine Liste der aktuell geladenen Dokumente.

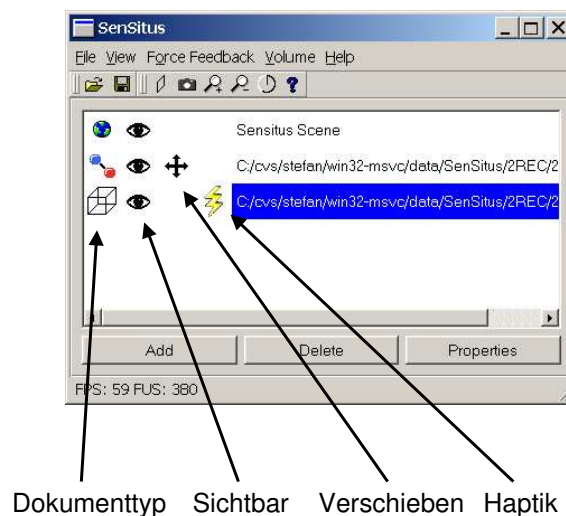


Abbildung 6.19: SenSitus: Hauptfenster

Eine Reihe von oft benötigten Aktionen können direkt in der Dokumentenliste ausgelöst werden. Zum Beispiel kann die Darstellung einzelner Dokumente abgeschaltet werden. In

der Liste kann ebenfalls das Dokument ausgewählt werden, das zur Zeit von dem Eingabegerät manipuliert werden soll (meist ist dies die Maus, in einer VR-Umgebung kann dies aber auch ein Eingabestift oder ähnliches sein).

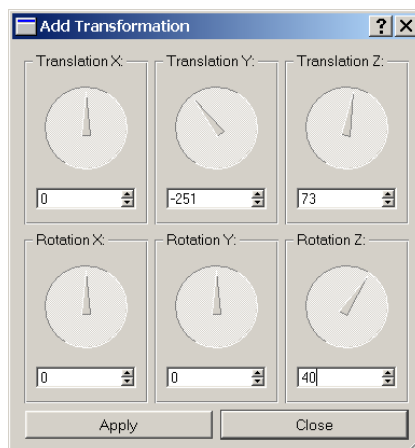


Abbildung 6.20: SenSitus: Transformationsdialog

Um die Dokumente mit einer höheren Präzision bewegen zu können, wurde eine Dialogbox entwickelt, die für alle sechs Freiheitsgrade Eingabewidgets zur Verfügung stellt (Abbildung 6.20). Die Translationen werden hierbei in 0,1 Ångström und die Rotationen in Grad angegeben. Um größere Strecken zurücklegen zu können, können die gerade eingestellten Werte in die Darstellung übernommen und die Drehregler wieder auf 0 gestellt werden.

Visualisieren von PDB-Dateien

Das Visualisieren von PDB-Dateien wurde schon in Kapitel 2.1.1 genauer vorgestellt. Eine der wichtigsten Eigenschaften für eine Visualisierung der Moleküle ist die Flexibilität der Darstellung. Alle Atome eines Moleküls in der gleichen Weise darzustellen ist oft nicht genug, gefordert sind Kombinationen von verschiedenen Darstellungsmodi.

Die zugrundeliegenden SVT-Klassen wurden schon beschrieben und erlauben beliebige Kombinationen der Darstellungsmodi. Es stellt sich die Frage, wie diese große Flexibilität auf eine einfach zu bedienende Benutzeroberfläche abgebildet werden kann. Einige Programme integrieren Skriptsprachen, mit denen die notwendigen Einstellungen vorgenommen werden können. So wird ein Maximum an Flexibilität erreicht, jedoch ist der Einsatz einer solchen Skriptsprache wenig intuitiv und mit einem hohen Lernaufwand verbunden.

Für das vorliegende Programm wurde eine Listen/Stack-Technik entwickelt, die gleichzeitig ein hohes Maß an Flexibilität und eine leicht erlernbare Benutzeroberfläche ermöglicht. Die Grund/-idee ist ein Stack, der eine Menge an Darstellungsänderungen enthält. Der Stack wird von oben nach unten abgearbeitet; dabei werden die enthaltenen Einstellungen durchgeführt. Jedes Element des Stacks ist eine Menge von Atomen, der ein Darstellungsmodus zugeordnet wird.

Der erste Eintrag des Stack ist meist ein Grundmodus, der zunächst allen Atomen zugeordnet wird, zum Beispiel die Liniendarstellung. Danach werden einzelne Atomgruppen durch die Zuordnung komplexerer Darstellungsmodi besonders hervorgehoben.

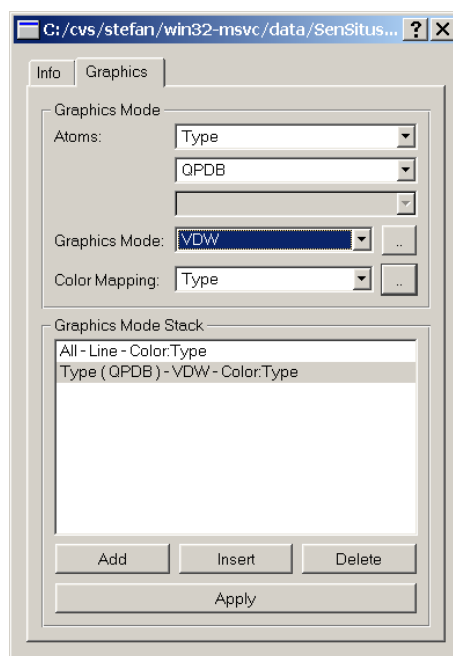


Abbildung 6.21: SenSitus: PDB-Einstellungsdialog

In Abbildung 6.21 ist der Einstellungsdialog dargestellt (hier “Graphics Mode Stack” genannt). Über das Widget “Atoms” kann für das selektierte Stackelement die Atommenge eingestellt werden, die dem Darstellungsmodus zugeordnet wird. Die Atommenge kann nach verschiedenen Kriterien selektiert werden:

Modus	Selektierte Atome
All	Alle Atome werden selektiert
Index	aus einem bestimmten Indexbereich
Type	eines bestimmten Typs
Residue Name	eines Residuums mit einem bestimmten Namen
Residue Seq.	eines Residuums mit einer bestimmten Sequenznummer
Chain ID	aus einer Kette mit einer bestimmten ID
Segment ID	aus einem Segment mit einer bestimmten ID

Bestimmte Selektionsmodi benötigen noch weitergehende Einstellungen. So kann für “Index” noch ein Intervall angegeben werden.

Wurde die Atommenge selektiert, so muss ihr noch ein Darstellungsmodus zugeordnet werden. Hier stehen die schon bei `svt_pdb` beschriebenen Modi (Abschnitt 6.2.4) zur Auswahl: Punkt-, Linien-, VDW-, CPK-, Dotted-, Licorice- und Tubedargestellung. Bei eini-

gen Darstellungsmodi können noch verschiedene Parameter beeinflusst werden, zum Beispiel zur Anpassung der Skalierung der Kugeln im VDW-Modus.

Die Atome können nach verschiedenen Kriterien eingefärbt werden. Naheliegender ist eine Farbgebung je nach Atomtyp. Es können obige Kriterien wie die Residuum- oder Segmentzugehörigkeit als Kriterium für die Einfärbung genutzt werden. Die verwendete Farbpalette und die Zuordnung der Farben zu den Kriterien kann ebenfalls editiert werden.

Visualisierung von Elektronenmikroskopie-Dichteverteilungen

Ähnlich wie für die gerade besprochene Visualisierung der Molekülstrukturdaten existiert auch für die Dichteverteilungen eine Dialogbox, in der alle notwendigen Einstellungen vorgenommen werden können.

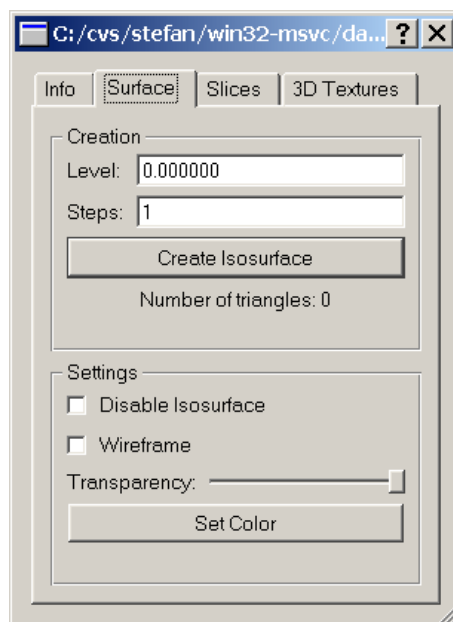


Abbildung 6.22: SenSitus: Einstellungsdialog Oberflächendarstellung

Die verschiedenen Möglichkeiten zur Visualisierung von Volumendaten, die im `svt_core` Modul implementiert wurden, können in SenSitus genutzt werden.

Zur Oberflächenvisualisierung wurde in `svt_core` der in Abschnitt 6.2.2 beschriebene Marching-Cube-Algorithmus implementiert. In Abbildung 6.22 ist die Dialogbox zur Erzeugung des Oberflächennetzes dargestellt. Der Marching-Cube-Algorithmus benötigt einen Schwellwert, durch den die Oberfläche innerhalb des Volumendatensatzes bestimmt wird. Dieser kann mit Hilfe des Widgets "Level" angegeben werden.

Die Dreiecksnetze, die der Algorithmus erzeugt, sind abhängig von dem zugrundeliegenden Datensatz recht groß. Um die Dreiecksmenge zu reduzieren, besteht die Möglichkeit,

die Schrittweite des Algorithmus zu erhöhen. So werden wesentlich weniger Dreiecke erzeugt, jedoch werden durch die größere Schrittweite kleine Details übersprungen, die dann auch in der Oberflächendarstellung nicht sichtbar werden.

Im Verlauf des Einpassprozesses werden die Molekülsstrukturdaten in die Dichteverteilungen bewegt. Um die Moleküle auch dann noch sichtbar darstellen zu können, muss das Oberflächennetz transparent dargestellt werden.

Ein echter Transparenzmodus ist in OpenGL nur rechenzeitintensiv realisierbar, da die Reihenfolge, in der die Dreiecke gezeichnet werden, nun wichtig ist. Alle zu zeichnenden Dreiecke müssen in Richtung der z -Achse sortiert werden, was jedoch bei den vielen Dreiecken der Dreiecksnetze einer Oberflächendarstellung zu aufwendig ist. Zudem müsste die Sortierung bei jeder Änderung der Szene wiederholt werden, was gerade bei den in VR-Umgebungen ständig wechselnden Betrachterpositionen nicht interaktiv umsetzbar ist.

Daher wurde für SenSitus eine Möglichkeit gesucht, um die Sortierung zu umgehen, und mit dem sogenannten “Stipple”-Modus gefunden. Hier wird von einem transparenten Dreieck nur jeder zweite (oder allgemeiner jedes n -te) Pixel gezeichnet. Bei den heute üblichen hohen Bildschirmauflösungen wirkt dies subjektiv ähnlich wie der Transparentmodus mit Dreieckssortierung.

Globale Darstellungseinstellungen

Nicht nur die einzelnen Dokumente sondern auch die gesamte dreidimensionale Szene kann beeinflusst werden. Zunächst einmal existiert in der globalen Dokumentenliste ein Szenenobjekt. Dieses Szenenobjekt bietet, wie die anderen Dokumente auch, die Möglichkeit, es zu verschieben und zu rotieren. Die Folgen bei dem globalen Szenenobjekt sind jedoch, dass alle Dokumente verschoben und rotiert werden. Dies ist sehr praktisch, wenn bei dem Einpassprozess schon eine gute Position für den Moleküldatensatz gefunden wurde. Über die globale Manipulation der Szene können Rotationen und Translationen ausgeführt werden, ohne eine Veränderung der relativen Position der Dokumente untereinander.

Daneben kann die Übersicht durch das Einfügen von Schnittebenen erhöht werden. Die Schnittebenen “zerschneiden” die dreidimensionale Darstellung. Es werden nur noch Punkte gezeichnet, die auf einer Seite der Schnittebene liegen, alle Punkte auf der anderen Seite werden unterdrückt. SenSitus bietet es an, diese Schnittebenen der Szene hinzuzufügen. Sie tauchen dann wie die Datensätze in der Dokumentenliste auf und können genau wie die anderen Dokumente verschoben und rotiert werden. Es ist ebenfalls möglich, mehr als nur eine Schnittebene hinzuzufügen. Das Limit liegt bei der von der vorliegenden OpenGL-Implementierung zur Verfügung gestellten Schnittebenen.

Interessante Ansichten können als Bild abgespeichert werden. Hierbei wird auf der Festplatte eine Datei im TGA-Dateiformat angelegt. TGA ist ein verlustfreies Bildformat und die Dateien werden mit einem Namen versehen, der durchnummeriert wird. So ist es möglich,

während eines Programmlaufs eine Reihe von Bildschirmhalten zu speichern, ohne dass diese immer wieder überschrieben werden.

Kapitel 7

Beispielanwendungen

Dieses Kapitel untersucht verschiedene Eigenschaften des entwickelten Ansatzes mit Hilfe von beispielhaften Anwendungen. Es wird der Frage nachgegangen, welchen Einfluss die Anzahl der Codebookvektoren auf die Präzision der Kraftberechnung hat, wie sich ein Absenken der Auflösung der Dichteverteilung auswirkt und mit welcher Geschwindigkeit die Kraftberechnung in der Praxis abläuft.

Zunächst wird die Frage untersucht, ob die implementierte Kraftberechnungsroutine den in Kapitel 4 vorgestellten Ansatz verwirklicht. Die Kraft soll den Anwender zu einer besseren Abbildungsposition leiten. In der vorliegenden Problemstellung wird die Abbildungsposition mit Hilfe des Korrelationskoeffizienten C bewertet, so dass der Kraftvektor in Richtung ∇C zeigen sollte.

Den einfachsten Anwendungsfall stellt ein einzelnes Atom dar, das in eine berechnete Elektronenmikroskopie-Dichteverteilung eingepasst wird. Zur Berechnung der Dichteverteilung wird das Atom mit Hilfe von trilineararer Interpolation auf ein dreidimensionales Gitter verteilt (siehe Abschnitt 4.3.2). Anschließend wird mit Hilfe des Gaussfilters der Volumendatensatz in seiner Auflösung auf 10 Ångström reduziert.

Beim interaktiven Einpassprozess bewegt der Anwender das Atom in den berechneten Volumendatensatz. Die Korrelation ist bei einem einzelnen Atom leicht zu berechnen. Die Ausgangsgleichung

$$C(R, \mathbf{T}) = \sum_{i=1}^k \rho_{em}(\mathbf{w}_i(R, \mathbf{T}))$$

vereinfacht sich in diesem Fall zu:

$$C(\mathbf{T}) = \rho_{em}(\mathbf{a} + \mathbf{T})$$

wobei \mathbf{a} das Atom und \mathbf{T} die Translation innerhalb des Volumendatensatzes darstellt. Die Rotation eines einzelnen Atoms um seinen Ursprung führt zu keiner Veränderung der Korrelation, so dass auf den Parameter R ganz verzichtet werden kann. Dies bedeutet, dass

sich die Korrelation unmittelbar aus dem Wert des Volumens an der Position des Atoms bestimmen lässt.

Die Werte des Volumens lassen sich, da nur ein einzelnes Atom betrachtet wird, aus der Gleichung des Gaussfilters entnehmen:

$$g(x) = e^{-\frac{1.5x^2}{\sigma_G^2}}$$

wobei hier $\sigma_G = 5$ ist ($2\sigma_G = 10$ gibt die Form der Gaussglocke und damit die erzeugte Auflösung in Ångström an). Dies führt zu:

$$g(x) = e^{-\frac{1.5}{25}x^2}$$

und der zugehörigen Ableitung:

$$\frac{d}{dx}g = g'(x) = -\frac{3}{25}x e^{-\frac{1.5}{25}x^2}$$

Die Position, in der die Dichteverteilung aus dem Atom berechnet worden ist, ist die perfekte Einpassposition für das Atom.

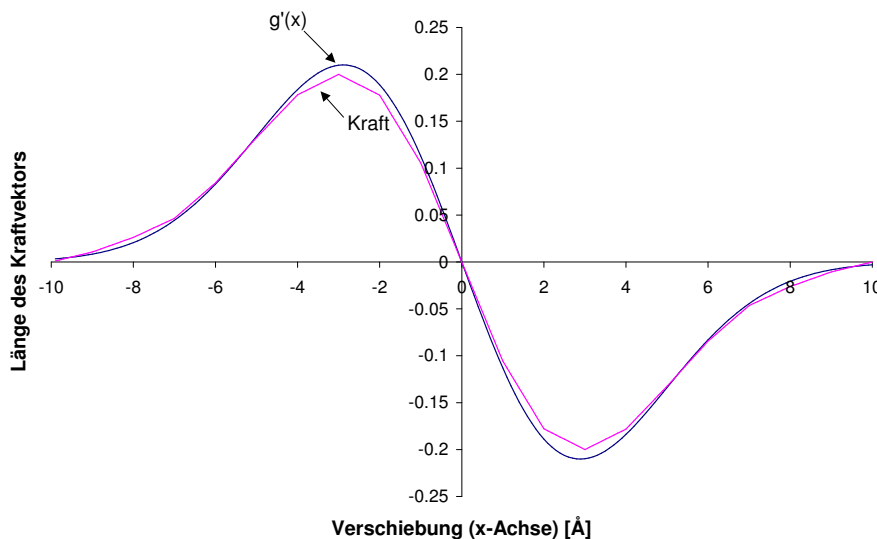


Abbildung 7.1: Kraftausgabe bei einem einzelnen Atom

Die Dichteverteilung und das Atom wurden daraufhin in das Programm SenSitus eingeladen und das Atom um -10 Ångström entlang der x-Achse verschoben. Anschliessend wurde die Kraftberechnung gestartet und das Atom mit einer Schrittweite von 0,1 Ångström so lange verschoben, bis die Position 10 Ångström erreicht war. Die Kraft wurde dabei aufgezeichnet, siehe Abbildung 7.1. Zum Vergleich wurde ebenfalls der Graph der Ableitung der Gaussfunktion dargestellt.

Es lässt sich unmittelbar der Zusammenhang zwischen der berechneten Kraft und der Ableitung der Gaussfunktion erkennen. Die geringen Unterschiede resultieren daraus, dass $g'(x)$ eine kontinuierliche Funktion ist, die Kraft jedoch aus einem diskreten Volumendatensatz berechnet wird. Es kann somit davon ausgegangen werden, dass die implementierte Kraftberechnung das theoretisch entwickelte Verfahren umsetzt.

Welchen Einfluss hat die Anzahl der Codebookvektoren?

Die vorgeschlagene Abbildungsmethode berechnet die auszugebende Kraft mit Hilfe von sogenannten Merkmalsvektoren (auch Codebookvektoren genannt), die durch Vektorquantisierung der hochaufgelösten Moleküldatensätze gewonnen werden. Letztlich stellt die Vektorquantisierung eine verlustbehaftete Komprimierungstechnik dar, so dass sich die Frage nach der Präzision der ausgegebenen Kraft stellt.

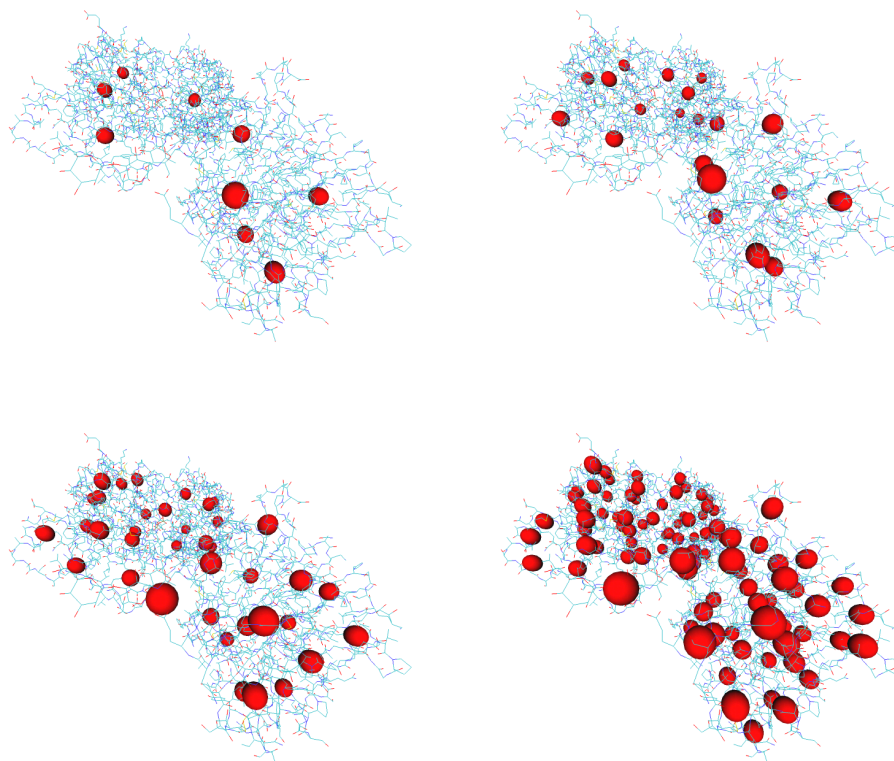


Abbildung 7.2: 10, 20, 40 und 100 Codebookvektoren - 1ATN

Die Präzision der Kraftberechnung lässt sich an zwei Eigenschaften festmachen. Zunächst sollte in der Nähe der perfekten Abbildungsposition ein Minimum der Kraftberechnung existieren. Je weiter das Kraftminimum von der besten Abbildungsposition entfernt ist, desto schlechter ist die Genauigkeit der über die Kraft transportierten Information.

Weiterhin sollte die Kraft innerhalb eines relativ kleinen Bereichs trichterförmig abfallen, um dem Anwender eine eindeutige Information zur Verfügung zu stellen. Fällt die Kraft innerhalb eines größeren Bereichs auf ein nahezu gleichbleibendes Niveau ab, so kann der Anwender hieraus keinen eindeutigen Hinweis auf eine gute Abbildungsposition ableiten.

In Abbildung 7.3 werden verschiedene Kraftberechnungen dargestellt, wobei analog zum vorherigen Abschnitt die hochaufgelösten Daten mit einer Schrittweite von 0,1 Ångström von -10 Ångström nach +10 Ångström (relativ zur perfekten Abbildungsposition) bewegt wurden.

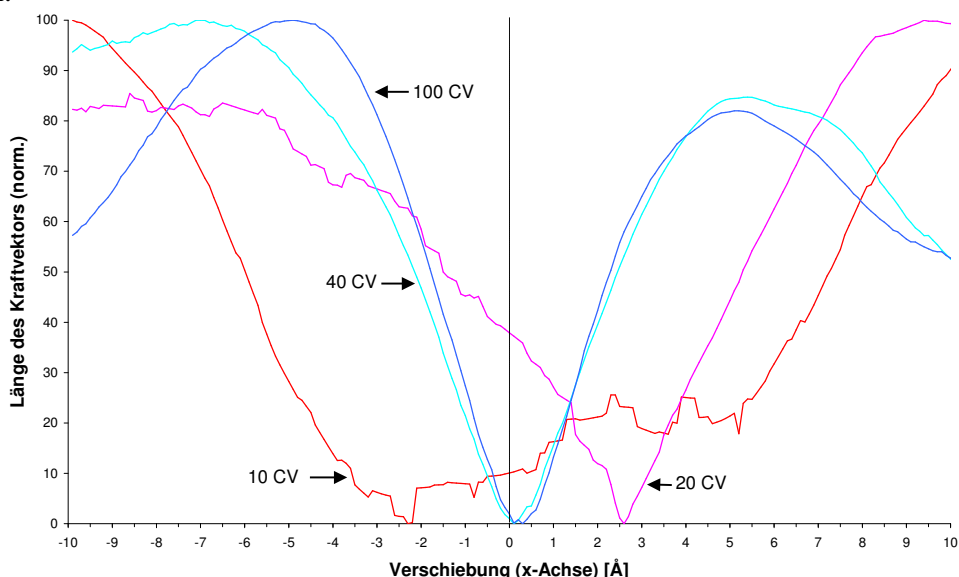


Abbildung 7.3: Kraftausgabe bei Verschiebung aus perfekter Abbildungsposition (1 ATN)

Es wurde das Molekül mit dem Eintrag 1 ATN aus der Protein-Data-Bank untersucht, welches aus 5020 Atomen besteht. Es wurde analog zum vorherigen Abschnitt aus dem Molekül eine Dichteverteilung mit 10 Ångström Auflösung berechnet.

Da die perfekte Abbildungsposition bekannt ist, sollte bei einer Verschiebung um 0 Ångström relativ zu dieser Position auch das Minimum der Kraftberechnung festgestellt werden können. Durch die verlustbehaftete Kompression ist jedoch eine geringe Abweichung zu vermuten.

Es wurde die Kraft für 10, 20, 40 und 100 Codebookvektoren (siehe Abbildung 7.2) aufgezeichnet und in Abbildung 7.3 dargestellt. Es ist zu erkennen, dass sowohl bei 10 als auch bei 20 Vektoren eine Abweichung des Kraftminimums vom Nullpunkt der Verschiebung um ungefähr 2,5 Ångström auftritt. Zudem ist bei 10 Codebookvektoren ein generelles Absinken der Kraft im Bereich von -7 bis +7 Ångström festzustellen. Dies bedeutet, dass die Kraftberechnung dem Anwender in diesem Fall nur eine relativ ungenaue Information über eine geeignete Abbildungsposition zur Verfügung stellt.

Bei 40 Vektoren ist das Minimum der berechneten Kraft exakt bei der perfekten Abbildungsposition zu erkennen, und zudem sinkt die Kraft in einem kleinen Bereich trichter-

förmig ab. Der Anwender kann aus der Kraftinformation also eindeutig eine gute Abbildungsposition ableiten. Im vorliegenden Beispiel entsteht bei 100 Codebookvektoren eine geringe Abweichung des Kraftminimums von zirka 0,2 Ångström. Ein weiterer Anstieg der Präzision ist also bei 100 Vektoren nicht festzustellen.

Festzuhalten bleibt, dass bei einer relativ geringen Anzahl von Codebookvektoren schon eine präzise Kraftinformation berechnet werden kann. In dem untersuchten Beispiel eines Moleküls von 5020 Atomen reichen schon 40 Vektoren aus, um ein Kraftminimum bei der besten Abbildungsposition zu berechnen. Ebenfalls ging aus dem Verlauf der Kraft ein eindeutiges Votum für eine geeignete Abbildungsposition hervor.

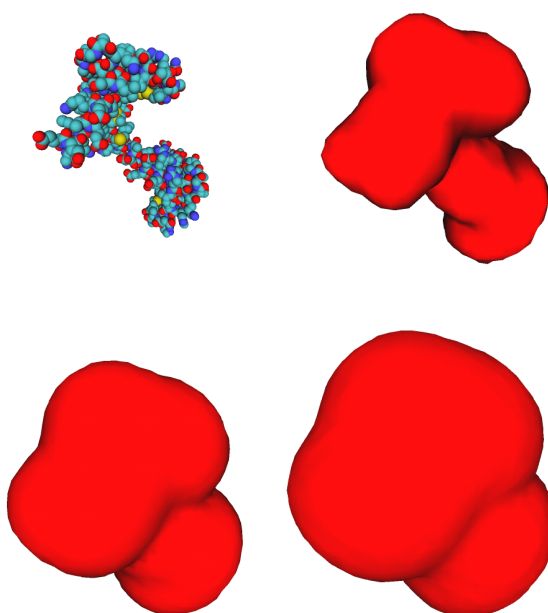


Abbildung 7.4: Molekül 1CLL und Dichteverteilungen (10Å, 20Å, 30Å Auflösung)

Wird ein weniger komplexes Molekül abgebildet, so verringert sich die benötigte Anzahl an Codebookvektoren. Das Molekül mit dem Eintrag 1CLL in der Protein-Data-Base ist in Abbildung 7.4 dargestellt und besitzt 1133 Atome. Es wurde die Kraftberechnung bei diesem Molekül ebenfalls untersucht (Abbildung 7.5).

Schon die Kraftausgabe mit 10 Codebookvektoren ist bei diesem Beispiel deutlich präziser als bei dem Molekül 1ATN. Es kann nur noch eine Abweichung von weniger als einem Ångström festgestellt werden. Ebenso ist ein Absinken der Kraft innerhalb eines relativ kleinen Bereichs feststellbar. Werden 20 Codebookvektoren verwendet, so entsteht nur noch eine Abweichung von zirka 0,3 Ångström, die auch bei 40 Codebookvektoren nicht mehr wesentlich kleiner wird. Für dieses deutlich kleinere Molekül reichen also schon 20 Codebookvektoren für eine präzise Kraftberechnung aus.

Analoge Feststellungen können auch für das Drehmoment getroffen werden. In Abbildung

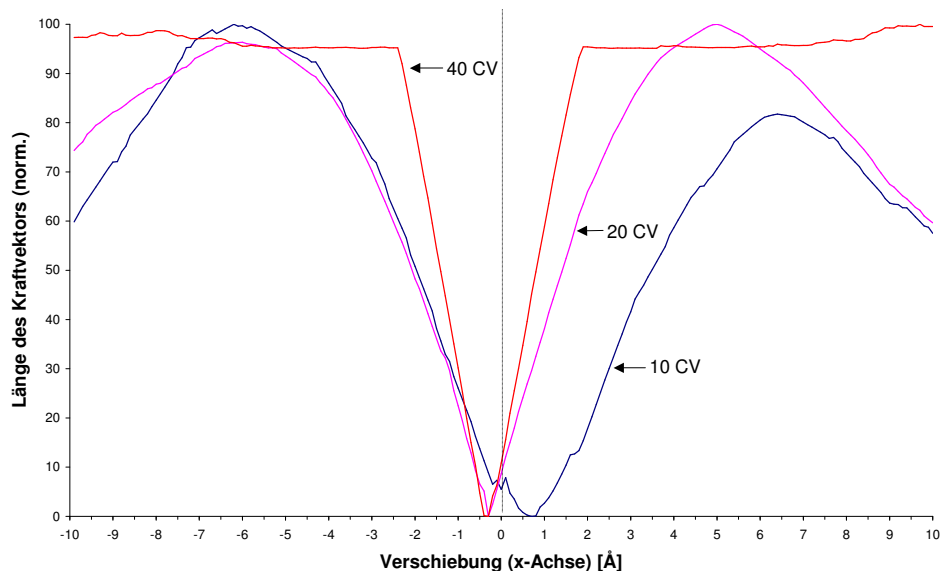


Abbildung 7.5: Kraftausgabe bei Verschiebung aus perfekter Abbildungsposition (1CLL)

7.6 wurde das Drehmoment bei Rotation des Moleküls 1CLL aus der perfekten Abbildungsposition dargestellt. Bei 10 Codebookvektoren erreicht das Drehmoment mit einer geringen Abweichung von der optimalen Orientierung sein Minimum, wohingegen sowohl bei 20 als auch bei 40 Vektoren die optimale Orientierung mit dem Minimum zusammenfällt.

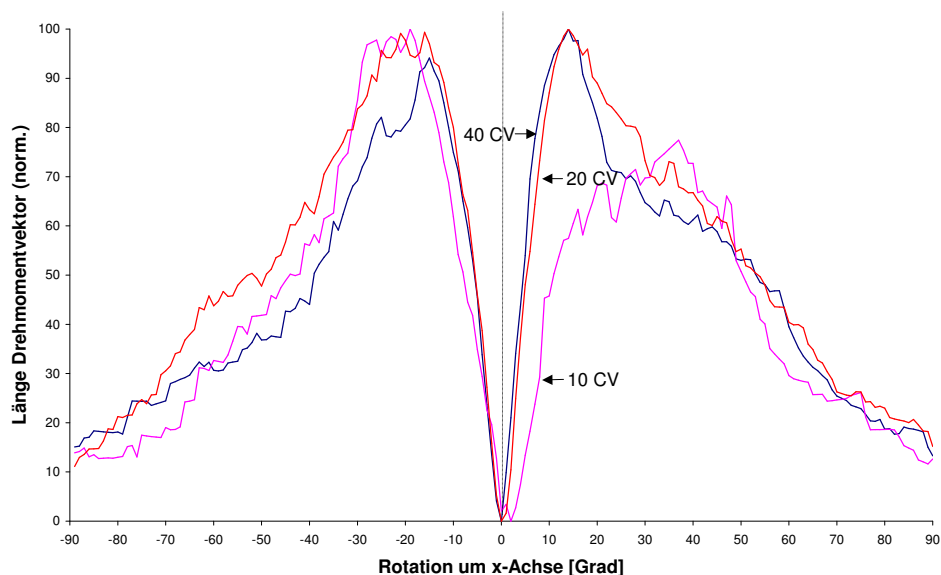


Abbildung 7.6: Kraftausgabe bei Rotation aus perfekter Abbildungsposition (1CLL)

Es bleibt festzuhalten, dass die Anzahl der Codebookvektoren einen wichtigen Einfluss auf die Präzision der Kraftberechnung hat. Wie gerade experimentell nachgewiesen wurde, reichen schon wenige Codebookvektoren aus, um wesentlich größere Moleküle mit einer

hohen Präzision mit Hilfe von Krafrückkopplung interaktiv einzupassen.

Bei der gerade vorgestellten Argumentation wurde die Ursache der auftretenden Ungenauigkeiten in der zu geringen Anzahl der Codebookvektoren gesucht. Gleichzeitig sollte jedoch nicht ignoriert werden, dass die Daten selbst eine perfekte Abbildung mit einer Abweichung des Kraftminimums von 0 Ångström nicht zulassen. Die Dichteverteilung hatte im gerade durchgeführten Test nur eine Auflösung von 10 Ångström und liegt zudem nicht kontinuierlich, sondern nur auf einem diskreten dreidimensionalen Gitter vor. Das Gitter teilt den Raum in Voxel ein, die eine Kantenlänge von 3 Ångström haben — innerhalb eines Voxel liegt überall nur der gleiche Dichtewert vor. Aus diesen Eigenschaften folgt, dass auch bei einer Kraftberechnung ohne Einsatz der Vektorquantisierungsmethode ein Fehler auftreten würde.

Gundsätzlich stellt sich die Frage nach einer optimalen Anzahl an Codebookvektoren für eine Molekülstruktur. Hierzu kann der vorgestellte Test auch automatisiert werden. Dazu wird eine zu unterschreitende Fehlerschranke vom Benutzer festgelegt. Aus der Molekülstruktur wird eine Dichteverteilung berechnet, deren Auflösung sich an der Elektronenmikroskopie-Dichteverteilung orientiert, in die die Struktur eingepasst werden soll.

Nun wird für die Molekülstruktur eine ansteigende Anzahl an Codebookvektoren berechnet. Jede Menge wird automatisch durch die gerade berechnete Dichteverteilung bewegt und die Kraft aufgezeichnet. Liegt die Abweichung des Kraftminimums unterhalb der angestrebten Fehlerschranke, so hat man eine möglichst geringe Codebookvektormenge gefunden, die die Molekülstruktur ausreichend gut beschreibt. Für den eigentlichen Einpassvorgang in die experimentelle Elektronenmikroskopie-Dichteverteilung kann diese Codebookvektormenge verwendet werden.

Welchen Einfluss hat die Auflösung der Dichteverteilung?

Wird die Auflösung der Dichteverteilung verringert, so kann auch ein Absinken der Präzision der Kraftberechnung vermutet werden. Die Dichteverteilung beinhaltet bei sinkender Auflösung immer weniger Information, die für ein Einpassen der hochauflösenden Molekülstruktur auf der Grundlage des Korrelationskoeffizienten notwendig ist.

Um dies zu untersuchen, wurde das Molekül 1CLL in Dichteverteilungen mit 10, 20 und 30 Ångström (siehe Abbildung 7.4) eingepasst. Wie im vorherigen Abschnitt wurde das Molekül aus der perfekten Abbildungsposition heraus um -90 bis 90 Grad in 1 Grad-Schritten rotiert.

Das Drehmoment wurde aufgezeichnet und in Abbildung 7.7 dargestellt. Es kann festgestellt werden, dass das minimale Drehmoment bei sinkender Auflösung von der optimalen Rotation (0 Grad) abweicht. Bei 30 Ångström ist die Abweichung jedoch gering (zirka 1 Grad Abweichung). Daneben fällt auf, dass die Kraft weniger schnell sinkt, je geringer die

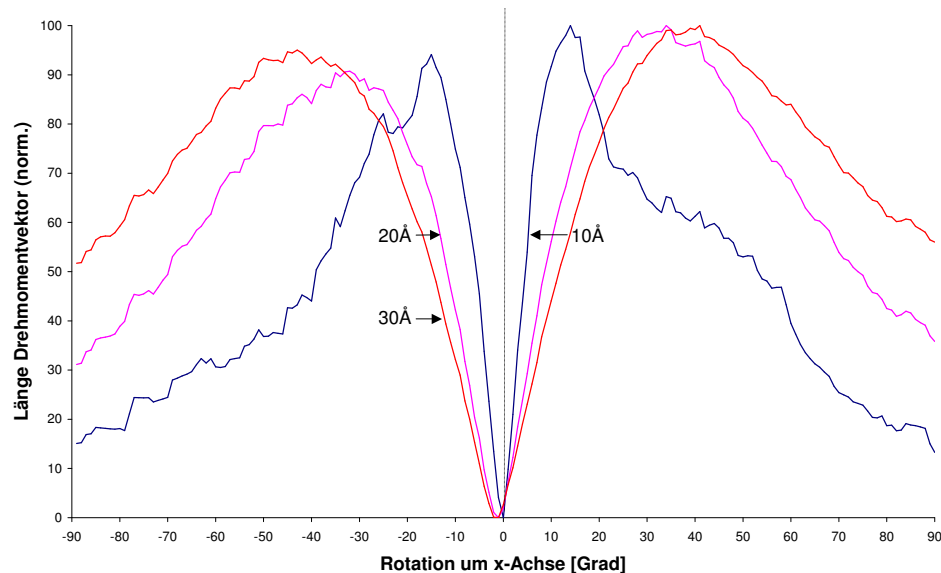


Abbildung 7.7: Kraftausgabe bei unterschiedlicher Auflösung der Dichteverteilung (1CLL)

Auflösung der Dichteverteilung ist. In keinem Fall kann aber ein Absinken auf ein niedriges Niveau über einen längeren Bereich festgestellt werden. Der Anwender erhält also eine eindeutige Information über eine bessere Abbildungsposition.

Wie schnell ist die Kraftberechnung?

Wie in Kapitel 3 beschrieben, wird in der Literatur davon ausgegangen, dass für das haptische Rendern sehr hohe Kraftneuberechnungsfrequenzen benötigt werden. Um einen realistischen Krafteindruck erreichen zu können, wird allgemein zirka 1000 Hertz als geeignete Berechnungsfrequenz angegeben [14].

Das entwickelte Programmsystem SenSitus wurde auf einem Einzelprozessor-PC unter Windows getestet, um eine typische Rechnerausstattung einzusetzen, die auch bei einem Forscher aus dem Bereich der strukturellen Biologie genutzt wird. Bei einem solchen System (1,13 GHz Taktfrequenz, Pentium III Prozessor, 256 MB RAM) können die in Abbildung 7.8 dargestellten Kraftberechnungsfrequenzen erreicht werden. Da die Frequenzen durch die Hintergrundaktivitäten des Betriebssystems beeinflusst werden, ist die Berechnungsrate nicht konstant. Als Mittelwert ergeben sich für das Molekül 1CLL:

Anzahl Vektoren	Mittelwert Kraftberechnungsrate
10	2530
20	1563
40	954

Wie oben festgestellt wurde, wird das Molekül 1CLL mit 20 Vektoren schon sehr gut angenähert. Die hier erreichte mittlere Kraftberechnungsfrequenz von 1563 liegt deutlich

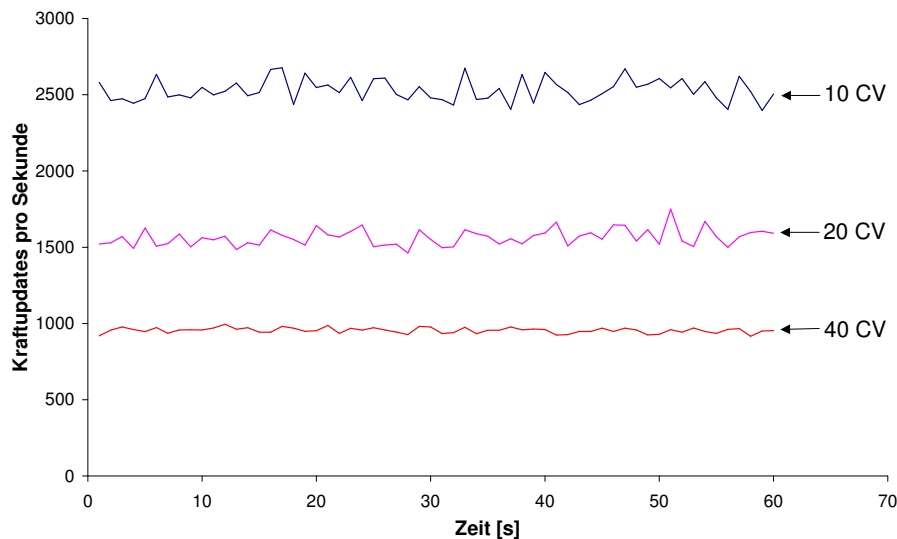


Abbildung 7.8: Geschwindigkeit der Kraftausgabe auf einem Single-CPU System

über den für das haptische Rendern oft genannten 1000 Hertz, die für einen realistischen Krafteindruck benötigt werden. Die vorgeschlagene Methode kann also selbst auf relativ leistungsschwachen PCs eingesetzt werden.

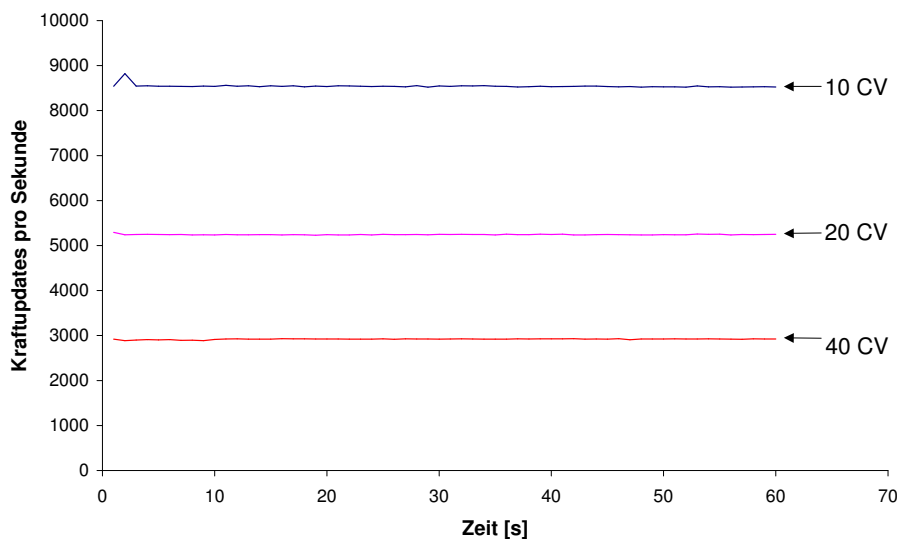


Abbildung 7.9: Geschwindigkeit der Kraftausgabe auf einem Dual-CPU System

Stehen leistungsfähigere Mehrprozessor-Systeme zur Verfügung, dann profitiert SenSitus von den zusätzlichen Prozessoren. SenSitus nutzt mehrere parallele Threads, so dass zum Beispiel die Kraftberechnung auf einen separaten Prozessor ausgelagert werden kann. Dies geschieht vollkommen automatisch durch das jeweils verwendete Betriebssystem. Getestet wurde das Programm mit identischen Datensätzen auf einer Dual-CPU-Workstation mit 1 Ghz AMD Athlon Prozessoren. Wie in Abbildung 7.9 zu erkennen ist, steigt die Geschwindigkeit der Kraftberechnung durch die Verwendung der zweiten CPU deutlich an.

Anzahl Vektoren	Mittelwert Kraftberechnungsrate
10	8542
20	5244
40	2919

Wie an den Mittelwerten abgelesen werden kann, liegen auf einem solchen System auch 40 Codebookvektoren mit 2919 Kraftberechnungen pro Sekunde deutlich über dem 1kHz-Kriterium für das haptische Rendern.

Unterstützt das haptische Rendern den Einpassprozess?

Es stellt sich generell die Frage, ob es sinnvoll ist, haptisches Rendern einzusetzen: Kann ein Anwender von der zusätzlichen Informationsquelle profitieren und seine Aufgabe schneller und präziser durchführen als ohne haptisches Rendern?

Es wurden in der Vergangenheit diverse Untersuchungen durchgeführt, um diese Frage zu beantworten. Burdea [12] beschreibt zum Beispiel ein Projekt von Hannaford und Wood, in dem Probanden sowohl mit als auch ohne haptisches Rendern einen Zapfen in eine Aufnahme einpassen sollten. Es zeigte sich, dass ohne haptische Information der Prozess des Einpassens nicht nur langsamer, sondern auch mit größerem Kraftaufwand und mit mehr Kollisionen ablief.

Das haptische Rendern zeigte sich in dieser und anderen Untersuchungen als wichtige Informationsquelle für den Benutzer. Dies ist auch nicht verwunderlich, da die sehr hohe zeitliche Auflösung des taktilen Wahrnehmungssystems des Menschen dem Gehirn bei einem interaktiven Prozess eine große Informationsmenge zur Verfügung stellt. Dies wird auch durch die erfolgreiche Anwendung von haptischem Rendern bei diversen Projekten [7, 14, 19, 27, 51] untermauert.

Auf eine systematische Evaluation des hier vorgestellten Systems wurde in der vorliegenden Arbeit bewusst verzichtet. Eine solche Evaluation hätte hier die Frage beantworten müssen, ob der Einsatz von haptischem Rendern auch in dieser Arbeit zu schnelleren und präziseren Ergebnissen führt. Hierzu hätten Probanden analog zu den gerade beschriebenen Untersuchungen ein hochauflösendes Molekül in eine niedrigauflösende Dichteverteilung sowohl mit als auch ohne haptisches Rendern einpassen müssen.

Für einen Nicht-Experten auf dem Gebiet der strukturellen Biologie hätte in diesem Fall aber nur die optische beziehungsweise haptische Information zur Verfügung gestanden, er hätte über kein zusätzliches Kontextwissen über die Datensätze verfügt. So wäre der Prozess des Einpassens für den Probanden vollkommen analog zu der Untersuchung von Hannaford und Wood verlaufen, nur dass der Zapfen und die Aufnahme eine andere Form besessen hätten. Eine Evaluation hätte vielmehr die Frage beantworten müssen, ob *Experten* mit der vorgestellten Lösung dank des haptischen Renderns schneller präzise Ergebnisse erzielen können. Da Experten der strukturellen Biologie jedoch nicht in einer größeren

Anzahl für eine statistisch aussagekräftige Untersuchung zur Verfügung standen, musste auf eine solche verzichtet werden.

Das System wurde jedoch auf der Konferenz “Biophysical Society - Discussions 2002” in Asilomar/USA vorgestellt [65]. Hier hatten die Teilnehmer — zumeist Experten der strukturellen Biologie — die Gelegenheit, das in dieser Arbeit vorgestellte System selbst an verschiedenen Datensätzen praktisch ausprobieren zu können. Eine mehrheitlich sehr positive Reaktion bestätigte die Erwartung, dass das haptische Rendern auch in der vorliegenden Anwendung sinnvoll eingesetzt werden kann.

Kapitel 8

Zusammenfassung und Ausblick

Die Arbeit hat eine neue interaktive Methode zum Abbilden von hochaufgelösten Molekülstrukturdaten in niedrigaufgelöste Elektronenmikroskopie-Dichteverteilungen vorgeschlagen. Es werden sowohl Virtual-Reality-Methoden als auch haptisches Rendern dazu genutzt, die Komplexität des Einpassprozesses für den Anwender zu reduzieren.

Das Abbilden von hochaufgelösten Molekülstrukturdaten in niedrigaufgelöste Dichteverteilungen ist ein seit vielen Jahren etabliertes Verfahren. Auf diese Weise können die Stärken unterschiedlicher Strukturermittlungstechniken miteinander kombiniert werden und so Aufnahmen erzeugt werden, die mit keiner Mikroskopietechnik alleine realisierbar sind.

Es wurden verschiedene Abbildungsverfahren beschrieben und verglichen. Es zeigte sich, dass sowohl die interaktiven Ansätze als auch die algorithmischen Programme Schwächen besitzen. Die interaktiven Programme verlagern die gesamte Komplexität des Abbildungsvorgangs auf den Anwender. Die algorithmischen Programme versuchen, das Problem vollautomatisch ohne Mitwirkung des Experten zu lösen. Durch die Eigenschaften der Daten (niedrige Auflösung, Rauschen, Defekte) sowie durch die Größe des Suchraumes wird die Aufgabenstellung für einen Algorithmus jedoch sehr komplex. Zudem ist eine Bewertung einer Abbildungssituation schwierig und mit den zur Zeit zur Verfügung stehenden Kriterien nur unzureichend gelöst.

Die vorliegende Arbeit schlägt daher einen interaktiven Ansatz vor, der die Stärken der etablierten Programme kombiniert. Es werden Methoden der algorithmischen Programme eingesetzt, um den Anwender während des interaktiven Einpassprozesses zu unterstützen. Dazu wird haptisches Rendern eingesetzt, das für den Benutzer eine weitere Informationsquelle darstellt.

Unter haptischem Rendern wird allgemein das künstliche Hervorrufen eines taktilen Sinneseindrucks verstanden. Der Begriff des taktilen Sinneseindrucks wird hier sehr umfassend definiert, da sowohl haptische und kinesthetische als auch propriozeptive Reize hierunter verstanden werden. Haptische Sinneseindrücke entstehen beim Abtasten einer Ober-

fläche und werden von Nervenzellen in der Haut wahrgenommen. Dagegen wird über kinesthetische Sinneseindrücke eine Kraft und über propriozeptive Eindrücke eine Positions- oder Lageveränderung wahrgenommen. Haptisches Rendern kann diese Sinneswahrnehmungen künstlich erzeugen, indem spezielle Eingabegeräte verwendet werden. In der vorliegenden Arbeit wurden Geräte der Firma SensAble eingesetzt. Mit Hilfe eines Phantom Premium 1.5 und eines Phantom 6DOF kann eine Kraft und ein Drehmoment auf die Hand eines Benutzers übertragen werden. Gleichzeitig sind die Geräte in der Lage, die Position und Orientierung der Hand zu bestimmen.

Der interaktive Einpassprozess wird vom Anwender mit Hilfe des haptischen Gerätes durchgeführt. Die aktuelle Position und Orientierung der Hand wird auf den hochaufgelösten Moleküldatensatz übertragen und so eine Abbildung in die niedrigaufgelöste Dichteverteilung hergestellt. Für die Kraftberechnung wurde mit dem Korrelationskoeffizienten eine Bewertungsfunktion der Abbildungssituation eingesetzt, die sehr häufig auch von algorithmischen Programmen genutzt wird. Der Anwender erhält so über die Kraftausgabe die Information, wie ein Algorithmus die Situation bewerten würde. Die Kraft wird so berechnet, dass der Anwender immer in Richtung einer ansteigenden Korrelation gezogen wird. Befindet er sich in einer sehr guten Abbildungsposition, so verspürt er daher keine Kraft mehr.

Das haptische Rendern benötigt für eine realistische Modellierung des Sinneseindrucks eine sehr hohe Kraftberechnungsfrequenz. Allgemein wird in der Literatur eine Frequenz von zirka 1000 Hertz für ausreichend gehalten. Bedingt durch die Größe der Volumendatensätze ist der Korrelationskoeffizient nur zeitaufwendig zu berechnen. Eine Kraftberechnung, die auf dieser Bewertungsfunktion basiert, wird eine Neuberechnungsfrequenz von 1000 Hertz in der Regel nicht erreichen können.

Zur Beschleunigung der Berechnungen wurde die Methode der Vektorquantisierung eingesetzt. Sie ermöglicht es, ein komplexes Signal mit Hilfe von wenigen Merkmalsvektoren zu beschreiben. Obwohl bei einer sehr geringen Anzahl von Vektoren das ursprüngliche Signal nicht mehr aus diesen herleitbar ist, beschreiben die Vektoren trotzdem die wichtigsten Eigenschaften. Wird dieses Verfahren auf die dreidimensionalen Volumendatensätze der vorliegenden Problemstellung angewendet, so beschreiben die Vektoren die dreidimensionale Struktur der Daten. Die große Datenmenge der Volumen kann so auf wenige Merkmalsvektoren komprimiert werden.

Als Vektorquantisierungsmethode wurden die Topology Representing Networks von Martinetz und Schulten eingesetzt. Diese Methode stellt einen Algorithmus aus dem Gebiet der Neuronalen Netzwerke dar. Es wird ein künstliches Neuronales Netzwerk so trainiert, dass die Gewichtsvektoren der Neuronen den Merkmalsvektoren entsprechen. Durch ein iteratives Verfahren wird das Netzwerk trainiert und so auf den jeweiligen Datensatz angepasst.

Mit Hilfe der erzeugten Merkmalsvektoren kann das hochaufgelöste Molekül kompakt beschrieben und daher die ansonsten aufwendige Korrelationsberechnung deutlich beschleunigt werden. Auf einer Mehrprozessorworkstation können auf diese Weise die hohen Kraft-

berechnungsraten von 1000 Hertz erreicht werden. Vorteilhaft ist, dass das Verfahren bezüglich der Berechnungsgeschwindigkeit unabhängig von der Größe der Volumendatensätze ist und nur noch von der Anzahl der Merkmalsvektoren abhängt. In Kapitel 7 konnte empirisch gezeigt werden, dass auch eine geringe Anzahl von Merkmalsvektoren zu einer relativ hohen Präzision der Kraftberechnung führt.

Da die Bewertungsfunktion im vorliegenden Ansatz gegenüber den algorithmischen Programmen interaktiv eingesetzt wird, können diverse Einschränkungen vermieden werden. Dies liegt vor allem daran, dass die vollautomatisch ablaufenden Programme sich nur auf die Korrelationsfunktion und die Datensätze zur Berechnung einer Lösung stützen können. In einem interaktiven Programm spielt das Kontextwissen des Anwenders eine wichtige Rolle. Dieser verfügt in der Regel noch über zusätzliches Wissen, zum Beispiel über die Entstehung der Datensätze. Bei der Suche nach einer geeigneten Abbildungsposition kann er dieses Wissen einbringen und kann daher viel leichter aus lokalen Maxima der Korrelationsfunktion das globale Maximum auswählen.

Die vorliegende Arbeit setzt neben dem haptischen Rendern auch Virtual-Reality-Methoden zur Unterstützung des Anwenders ein. Ein Virtual-Reality-System (kurz VR-System) ermöglicht es dem Benutzer, mit virtuellen Objekten ähnlich wie mit realen Objekten zu interagieren. Die virtuellen Objekte besitzen eine reale Raumposition und können daher wesentlich leichter dreidimensional erfasst werden. Dies ist in der vorliegenden Problemstellung von Bedeutung, sollen doch hier dreidimensionale Datensätze im Raum positioniert werden. Durch den Einsatz von VR-Systemen können die Lage, die Position und die dreidimensionalen Eigenschaften der zu positionierenden Objekte deutlich besser erfasst werden. In der vorliegenden Arbeit wurde das VR-System "Holobench" genutzt, das am Zentralinstitut für Angewandte Mathematik (ZAM) im Forschungszentrum Jülich installiert ist.

Der im Verlauf der Arbeit entwickelte Lösungsvorschlag wurde in Gestalt des Programms "SenSitus" implementiert. Das Programm ist portabel und unterstützt sowohl Shared-Memory-Parallelrechner unter diversen Unix-Betriebssystemen als auch PCs unter Linux und Windows. Es kann in VR-Umgebungen als auch auf normalen Workstations eingesetzt werden.

Als Grundlage für das Programm wurde ein System von Bibliotheken zur Ansteuerung von VR-Systemen und zur Visualisierung entwickelt. Mit Hilfe des "Scientific Visualization Toolkit" (kurz SVT) genannten Systems können Programme für VR-Umgebungen realisiert werden. Es wird nicht nur das am ZAM vorhandene Holobench-System unterstützt sondern praktisch beliebige Mehrbildschirm-Projektionssysteme. Eine ebenfalls im Rahmen der Arbeit entstandene Bibliothek mit dem Namen L.I.V.E. ("Library for Input-Devices in VR-Environments") übernimmt die Ansteuerung von Eingabegeräten. Durch das modulare Design lassen sich leicht Ansteuerrouninen für neue Geräte integrieren, sowie Eingabegeräte auch über das Netzwerk ansprechen. Auch haptische Geräte, wie die für die Arbeit verwendeten Geräte der Firma SensAble, werden unterstützt.

Für die Zukunft sind verschiedene Weiterentwicklungen der vorgestellten Methode denkbar. Vor allem könnte das realisierte “Rigid-Body-Docking” durch ein “Flexible-Fitting” ergänzt werden. Beim Rigid-Body-Docking-Verfahren wird davon ausgegangen, dass das hochaufgelöste Molekül wie ein starrer Körper in die niedrigaufgelöste Dichteverteilung abgebildet wird. Da jedoch beide Datensätze aus verschiedenen Experimenten stammen, wird sich der Zustand des hochaufgelösten Moleküls meistens von dem Zustand der niedrigaufgelösten Struktur unterscheiden. Das Flexible-Fitting verändert daher die hochaufgelöste Struktur während des Einpassprozesses. Dies führt aufgrund der vielen möglichen Zustandsänderungen zu einer stark erhöhten Menge an Freiheitsgraden, so dass nur sehr wenige algorithmische Programme dieses Verfahren umsetzen.

In einem interaktiven Programm könnte das haptische Rendern zur Manipulation des Zustands des Moleküls eingesetzt werden. Es könnten die biophysikalischen Grenzen der Verformung für eine Kraftberechnung genutzt werden. Der Anwender würde auf diese Weise darüber informiert, wie realistisch die gerade eingestellte Zustandsänderung ist. In diesem Zusammenhang könnten zur Reduktion der Komplexität wieder die “Topology Representing Networks” zum Einsatz kommen. Analog zu Wriggers Ansatz in [64] könnte das ermittelte Verbindungsnetzwerk zur Verformung der Strukturdaten eingesetzt werden.

Ebenso ist eine Integration von vollautomatisch ablaufenden Algorithmen in das Programm SenSitus möglich. Durch die Interaktionsmöglichkeiten des Programms könnten in einem Vorverarbeitungsschritt die Datensätze manipuliert und so der Suchraum für den Algorithmus eingeschränkt werden. Durch eine geeignete visuelle Rückkopplung könnte zudem der Algorithmus zur Laufzeit beobachtet und die aktuellen Maxima der Bewertungsfunktion, also die besten Abbildungspositionen angezeigt werden. So könnte der Algorithmus bei einem guten Ergebnis auch schon vor Ende der Berechnungen abgebrochen und Rechenzeit eingespart werden.

Literaturverzeichnis

- [1] AGRAWALA, Maneesh ; BEERS, Andrew C. ; MCDOWALL, Ian ; FRÖHLICH, Bernd ; BOLAS, Mark ; HANRAHAN, Pat: The Two-User Responsive Workbench: Support for Collaboration Through Independent Views of a Shared Space. In: *ACM Computer Graphics (SIGGRAPH '97 Proceedings)* Bd. 31, 1997, S. 327–332
- [2] ALBERTS, B.: The Cell as a Collection of Protein Machines: Preparing the Next Generation of Molecular Biologists. In: *Cell* 92 (1998), S. 291–294
- [3] ALLEN, M. P. ; TILDESLEY, D. J.: *Computer Simulations of Liquids*. Oxford University Press, 1987
- [4] AVS ADVANCED VISUAL SYSTEMS. *AVS/Express*.
URL <http://www.avs.com>
- [5] BELITZ, M.: Aufbau eines Analog-/Digitalwandler-Interface für Eingabegeräte an VR-Systemen / Zentralinstitut für Angewandte Mathematik, Forschungszentrum Jülich. 2000. – Studienarbeit
- [6] BELNAP, David M. ; KUMAR, Abhinav ; FOLK, Jon T. ; SMITH, Thomas J. ; BAKER, Timothy S.: Low-Resolution Density Maps from Atomic Models: How Stepping Back Can Be a Step Forward. In: *J. Struct. Biol.* 125 (1999), S. 166–175
- [7] BERGMAN, Lawrence D. ; RICHARDSON, Jane S. ; RICHARDSON, David S. ; BROOKS, Frederick: VIEW - An Exploratory Molecular Visualization System with User-Definable Interaction Sequences. In: *ACM Computer Graphics (SIGGRAPH '93 Proceedings)* Bd. 27, 1993, S. 117–126
- [8] BERMAN, H. ; WESTBROOK, J. ; FENG, Z. ; GILLILAND, G. ; BHAT, T. ; WEISSIG, H. ; SHINDYALOV, I. ; BOURNE, P.: The Protein Data Bank. In: *Nucl. Acid. Res.* 28 (2000), S. 235–242
- [9] BERNSTEIN, F.C. ; KOETZLE, T.F. ; WILLIAMS, G. J. B. ; MEYER, E. F. J. ; BRICE, M.D. ; RODGERS, J. R. ; KENNARD, O. ; SHIMANOUCI, T. ; TASUMI, M.: The Protein Data Bank: A Computer-based Archival File for Macromolecular Structures. In: *J. Mol. Biol.* 112 (1977), S. 535–542
- [10] BIERBAUM, A.: *VR Juggler: A Virtual Platform for Virtual Reality Application Development.*, Iowa State University, Diplomarbeit, 2000

- [11] BROOKS, Frederick P.: What's Real About Virtual Reality? In: *IEEE Computer Graphics and Applications* 19 (1999), Nr. 6, S. 16–27
- [12] BURDEA, Grigore C.: *Force and Touch Feedback for Virtual Reality*. Wiley and Sons, 1996
- [13] CHACÓN, Pablo ; WRIGGERS, Willy: Multi-Resolution Contour-based Fitting of Macromolecular Structures. In: *J. Mol. Biol.* 317 (2002), S. 375–384
- [14] CHEN, E. ; B., Marcus: Force Feedback for Surgical Simulation. In: *Proceedings of the IEEE* 86(3) (1998), S. 524–530
- [15] COWTAN, Kevin: Modified Phased Translation Functions and their Application to Molecular-Fragment Location. In: *Acta Cryst. D* 54 (1998), S. 750–756
- [16] CRUZ-NEIRA, Carolina ; SANDIN, Daniel J. ; DEFANTI, Thomas A.: Surround-Screen Projection-Based Virtual Reality: The design and Implementation of the CAVE. In: *ACM Computer Graphics (SIGGRAPH '93 Proceedings)* Bd. 27, 1993, S. 135–142
- [17] FOLEY, James D. ; DAM, Andreis von ; FEINER, Steven K. ; HUGHES, John F.: *Computer Graphics - Principles and Practice*. Addison-Wesley, 1982
- [18] GUEx, N. ; PEITSCH, M.C.: SWISS-MODEL and the Swiss-PdbViewer: An Environment for Comparative Protein Modeling. In: *Electrophoresis* 18 (1997), S. 2714–2723
- [19] GUTHOLD, M. ; FALVO, M. ; MATTHEWS, W. G. ; PAULSON, S. ; MULLIN, J. ; LORD, S. ; D., Erie ; WASHBURN, S. ; SUPERFINE, R. ; BROOKS, F. P. ; TAYLOR, R. M.: Investigation and modification of molecular structures with the nanoManipulator. In: *Journal of Molecular Graphics and Modelling* 17 (1999), S. 187–197
- [20] HAENSLER, Eberhard: *Grundlagen der Theorie statistischer Signale*. Springer-Verlag, 1983
- [21] HAYKIN, S.: *Neural Networks - a comprehensive foundation*. Macmillan College Publishing Company, Inc., 1994
- [22] HAYWARD, V. ; ASTLEY, O. *Performance Measures for Haptic Interfaces*. 1996
- [23] HERTZ, John ; KROGH, Anders ; PALMER, Richard G.: *Introduction to the theory of neural computation*. Wylde, Allan M., Addison-Wesley Publishing Company, 1991 (A lecture notes volume in the Santa Fe Institute studies in the science of complexity)
- [24] HOPPE, Hugues: Progressive meshes. In: *Proceedings of the 23rd annual conference on computer graphics and interactive techniques*, ACM Press, 1996, S. 99–108
- [25] HUMPHREY, W. ; DALKE, A. ; SCHULTEN, K.: VMD - Visual Molecular Dynamics. In: *J. Mol. Graph.* 14 (1996), S. 33–38

- [26] HUMPHREYS, G. ; ELDRIDGE, M. ; BUCK, I. ; STOLL, G. ; EVERETT, M. ; HANRAHAN, P.: WireGL: A Scalable Graphics System for Clusters. In: *ACM Computer Graphics (SIGGRAPH '01 Proceedings)* Bd. 35, 2001, S. 129–140
- [27] IX, F. D. ; QIN, H. ; KAUFMAN, A. ; EL-SANA, J.: Haptic Sculpting of Dynamic Surfaces. In: *ACM Symposium on Interactive 3D Graphics*, 1999, S. 103–110
- [28] JONES, T. A. ; ZOU, J.-Y. ; COWAN, S. W.: Improved Methods for Building Protein Models in Electron Density Maps and the Location of Errors in these Models. In: *Acta Cryst. A* 47 (1991), S. 110–119
- [29] JONES, T. A.: A Graphics Model Building and Refinement System for Macromolecules. In: *J. Appl. Cryst.* 11 (1978), S. 268–272
- [30] KALE, L. ; SKEEL, R. ; BHANDARKAR, M. ; BRUNNER, R. ; GURSOY, A. ; KRAWETZ, N. ; PHILLIPS, J. ; SHINOZAKI, A. ; VARADARAJAN, K. ; SCHULTEN, K.: NAMD2: Greater Scalability for Parallel Molecular Dynamics. In: *Journal of Computational Physics* 151 (1999), S. 283–312
- [31] KLEYWEGT, Gerard J. ; JONES, Alwyn T.: Template Convolution to Enhance or Detect Structural Features in Macromolecular Electron-Density Maps. In: *Acta Cryst. D* 53 (1997), S. 179–185
- [32] KNISS, Joe ; KINDLMANN, Gordon ; HANSEN, Charles: Interactive Volume Rendering Using Multi-Dimensional Transfer Functions and Direct Manipulation Widgets. In: *Proceedings of Visualization 2001*, IEEE Computer Society, 2001, S. 255–262
- [33] LACROUTE, Philippe ; LEVOY, Marc: Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation. In: *ACM Computer Graphics (SIGGRAPH '94 Proceedings)* Bd. 28, 1994, S. 451–458
- [34] LINDE, Y. ; BUZO, A. ; GRAY, R. M.: An Algorithm for Vector Quantizer Design. In: *IEEE Transactions on Communications* (1980), January, S. 702–710
- [35] LORENSEN, William E. ; CLINE, Harvey E.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In: STONE, M. C. (Hrsg.): *ACM Computer Graphics (SIGGRAPH '87 Proceedings)* Bd. 21(4), 1987, S. 163–169
- [36] MARTINETZ, T. ; SCHULTEN, K.: Topology Representing Networks. In: *Neural Networks* 7 (1993), S. 507–522
- [37] MASSIE, T. ; SALISBURY, J.: The Phantom Haptic Interface: A Device for Probing Virtual Objects. In: *ASME Winter Annual Meeting, Symposium on Haptic Interfaces for a virtual environment and teleoperator systems*, 1994
- [38] MCCORMICK, B.H. ; DEFANTI, T.A. ; BROWN, M.D.: Visualization in Scientific Computing. In: *Computer Graphics* 21 (1987), Nr. 6
- [39] MINSKY, Margaret ; MING, Ouh-young ; STEELE, Oliver ; BROOKS, Frederick P. ; BEHENSKY, Max: Feeling and Seeing: Issues in Force Display. In: *Computer Graphics* 24 (1990), Nr. 2, S. 235–243

- [40] OPENAL. *OpenAL - Open Audio Library*.
URL <http://www.openal.org>
- [41] PFISTER, Hanspeter ; HARDENBERGH, Jan ; KNITTEL, Jim ; LAUER, Hugh ; SEILER, Larry: The VolumePro Real-Time Ray-Casting System. In: *ACM Computer Graphics (SIGGRAPH '99 Proceedings)* Bd. 33, 1999, S. 251–260
- [42] RADERMACHER, Paul. *GLUI User Interface Library*.
URL <http://www.cs.unc.edu/rademach/glui>
- [43] REIMERSDAHL, T. van ; KUHLEN, T. ; GERNDT, A. ; HENRICHS, J. ; BISCHOF, C.: ViSTA: a multimodal, platform-independent VR-Toolkit based on WTK, VTK, and MPI. In: *Proceedings of IPT 2000*, 2000 (Fourth International Immersive Projection Technology Workshop, Ames, Iowa)
- [44] RICHARDS, Frederic M.: The matching of physical models to three-dimensional electron-density maps: A simple optical device. In: *J. Mol. Biol.* 37 (1968), S. 225–230
- [45] ROSEMAN, Alan M.: Docking structures of domains into maps from cryo-electron microscopy using local correlation. In: *Acta Cryst. D* 56 (2000), S. 1332–1340
- [46] ROSSMAN, Michael G.: Fitting atomic models into electron-microscopy maps. In: *Acta Cryst. D* 56 (2000), S. 1341–1349
- [47] SAYLE, R. A. ; MILNER-WHITE, E. J.: RasMol: Biomolecular Graphics for all. In: *Trends Biochem. Sci.* 20 (1995), S. 374–376
- [48] SCHULTEN, Klaus. *The Theoretical Biophysics Group, Beckman Institute, University of Illinois at Urbana-Champaign*. URL <http://ks.uiuc.edu>
- [49] SENSABLE CORPORATION. *PHANToM Force-Feedback Devices*.
URL <http://www.sensable.com>
- [50] SRINIVASAN, M. A. ; CHEN, J.: Human performance in controlling normal forces of contact with rigid objects. In: *Advances in Robotics, Mechatronics and Haptic Interfaces* 49 (1993), S. 119–125
- [51] STONE, J. ; GULLINGSRUD, J. ; GRAYSON, P. ; SCHULTEN, K.: A System for Interactive Molecular Dynamics Simulation. In: HUGES, John F. (Hrsg.) ; SEQUIN, Carlo H. (Hrsg.): *ACM Symposium on Interactive 3D Graphics*, 2001, S. 191–194
- [52] SUTHERLAND, Ivan E.: The Ultimate Display. In: *Proceedings of the IFIP Congress* 65, 1965, S. 506–508
- [53] TAYLOR, Russell M. ; HUDSON, Thomas C. ; SEEGER, Adam ; WEBER, Hans ; JULIANO, Jeffrey ; HELSER, Aron T.: VRPN: A Device Independent, Network-Transparent VR Peripheral System. In: *Proceedings of the VRST 2001*, 2001 (ACM Symposium on Virtual Reality Software and Technology, Banff, Canada)

- [54] TAYLOR, Russell M. ; ROBINETT, Warren ; CHI, Vernon L. ; BROOKS, Frederick P. ; WRIGHT, William V. ; WILLIAMS, R. S. ; SNYDER, Erik J.: The Nanomanipulator: A Virtual-Reality Interface for a Scanning Tunneling Microscope. In: *ACM Computer Graphics (SIGGRAPH '93 Proceedings)* Bd. 27, 1993, S. 127–134
- [55] TAYLOR II., R.M. *VRPN: Virtual Reality Pheriphal Network*.
URL <http://www.cs.unc.edu/Research/vrpn>. 1998
- [56] TERARECON CORPORATION. *VolumePro Volume-Rendering PCI Accelerator*.
URL <http://www.terarecon.com>
- [57] TROLLTECH CORPORATION. *Graphical User Interface Library Qt*
- [58] VERLET, L.: Computer experiments on classical fluids: I. Thermodynamical properties of Lennard-Jones molecules. In: *Physical Review* 159 (1967), S. 98–103
- [59] VOLKMANN, Niels: Atomic Model of the Cell: Docking in a tomographic environment. In: *Proceedings of the Biophysical Society Discussions* Biophysical Society, 2002, S. SP11A–SP11E
- [60] VOLKMANN, Niels ; HANEIN, Dorit: Quantitative Fitting of Atomic Models into Observed Densities by Electron Microscopy. In: *J. of Struct. Biol.* 125 (1999), S. 176–184
- [61] VRCO CORPORATION. *CAVELib*.
URL <http://www.vrco.com/products/cavelib/cavelib.html>
- [62] WESTERMANN, R. ; SEVENICH, B.: Accelerated Volume Ray-Casting using Texture Mapping. In: *Proceedings of the Visualization 2001* IEEE Visualization, 2001, S. 271–278
- [63] WILSON, O. ; GELDER, A. V. ; WILHEMS, J.: Direct Volume Rendering via 3D Textures / University of California, Santa Cruz. 1994 (UCSC-CRL-94-19). – Forschungsbericht
- [64] WRIGGERS, W. ; BIRMANNS, S.: Using Situs for Flexible and Rigid-Body Fitting of Multiresolution Single-Molecule Data. In: *J. Struct. Biol.* 133 (2001), S. 193–202
- [65] WRIGGERS, W. ; BIRMANNS, S. ; TAMA, F. ; KOVACS, J. ; CHACÓN, P.: Reconciling Shape with Structure: Morphometric Strategies for Multi-Resolution Flexing. In: *Proceedings of the Biophysical Society Discussions* Biophysical Society, 2002, S. SP12A–SP12G
- [66] WRIGGERS, W. ; CHACÓN, P.: Modeling Tricks and Fitting Techniques for Multiresolution Structures. In: *Structure* 9 (2001), S. 779–788
- [67] WRIGGERS, Willy ; AGRAWAL, Radjendra K. ; DREW, Devin L. ; MCCAMMON, Andrew ; FRANK, Joachim: Domain Motions of EF-G Bound to the 70S Ribosome: Insights from a Hand-Shaking between Multi-Resolution Structures. In: *Biophys. Journal* 79 (2000), S. 1670–1678

- [68] WRIGGERS, Willy ; MILLIGAN, Ronald A. ; MCCAMMON, J. A.: Situs: A Package for Docking Crystal Structures into Low-Resolution Maps from Electron Microscopy. In: *J. of Struct. Biol.* 125 (1999), S. 185–195
- [69] WRIGGERS, Willy ; MILLIGAN, Ronald A. ; SCHULTEN, Klaus ; MCCAMMON, J. A.: Self-organizing Neural Networks Bridge the Biomolecular Resolution Gap. In: *J. Mol. Biol.* 284 (1998), S. 1247–1254

Lebenslauf

1.6.1972	Geboren in Aachen
1978-1982	Grundschule Klinkheide, Herzogenrath
1982-1991	Gymnasium der Stadt Herzogenrath - Abitur 1991
1991-1992	Zivildienst bei der Kindertagesstätte der Rheinisch-Westfälischen Technischen Hochschule Aachen
1992-1999	Studium der Informatik an der Rheinisch-Westfälischen Technischen Hochschule Aachen - Diplom 1999
seit 1999	Doktorand bei Prof. Friedel Hoßfeld am Zentralinstitut für Angewandte Mathematik Forschungszentrum Jülich
2000	Zweimonatiger Forschungsaufenthalt am Theoretical Biophysics Group, Prof. Klaus Schulten, Ph.D. Beckman Institute, University of Illinois at Urbana-Champaign, U.S.A
2001	Zweimonatiger Forschungsaufenthalt bei der Computational Structural Biology Group, Prof. Willy Wriggers, Ph.D. The Scripps Research Center, San Diego, U.S.A.

Bisher sind erschienen:

**Modern Methods and Algorithms of Quantum Chemistry -
Proceedings**

Johannes Grotendorst (Hrsg.)

NIC-Serie Band 1

Winterschule, 21. - 25. Februar 2000, Forschungszentrum Jülich

ISBN 3-00-005618-1, Februar 2000, 562 Seiten

nicht mehr lieferbar

**Modern Methods and Algorithms of Quantum Chemistry -
Poster Presentations**

Johannes Grotendorst (Hrsg.)

NIC-Serie Band 2

Winterschule, 21. - 25. Februar 2000, Forschungszentrum Jülich

ISBN 3-00-005746-3, Februar 2000, 77 Seiten

nicht mehr lieferbar

**Modern Methods and Algorithms of Quantum Chemistry -
Proceedings, Second Edition**

Johannes Grotendorst (Hrsg.)

NIC-Serie Band 3

Winterschule, 21. - 25. Februar 2000, Forschungszentrum Jülich

ISBN 3-00-005834-6, Dezember 2000, 638 Seiten

**Nichtlineare Analyse raum-zeitlicher Aspekte der
hirnelektrischen Aktivität von Epilepsiepatienten**

Jochen Arnold

NIC-Serie Band 4

ISBN 3-00-006221-1, September 2000, 120 Seiten

**Elektron-Elektron-Wechselwirkung in Halbleitern:
Von hochkorrelierten kohärenten Anfangszuständen
zu inkohärentem Transport**

Reinhold Löwenich

NIC-Serie Band 5

ISBN 3-00-006329-3, August 2000, 145 Seiten

**Erkennung von Nichtlinearitäten und
wechselseitigen Abhängigkeiten in Zeitreihen**

Andreas Schmitz

NIC-Serie Band 6

ISBN 3-00-007871-1, Mai 2001, 142 Seiten

**Multiparadigm Programming with Object-Oriented Languages -
Proceedings**

Kei Davis, Yannis Smaragdakis, Jörg Striegnitz (Hrsg.)

NIC-Serie Band 7

Workshop MPOOL, 18. Mai 2001, Budapest

ISBN 3-00-007968-8, Juni 2001, 160 Seiten

**Europhysics Conference on Computational Physics -
Book of Abstracts**

Friedel Hossfeld, Kurt Binder (Hrsg.)

NIC-Serie Band 8

Konferenz, 5. - 8. September 2001, Aachen

ISBN 3-00-008236-0, September 2001, 500 Seiten

NIC Symposium 2001 - Proceedings

Horst Rollnik, Dietrich Wolf (Hrsg.)

NIC-Serie Band 9

Symposium, 5. - 6. Dezember 2001, Forschungszentrum Jülich

ISBN 3-00-009055-X, Mai 2002, 514 Seiten

**Quantum Simulations of Complex Many-Body Systems:
From Theory to Algorithms - Lecture Notes**

Johannes Grotendorst, Dominik Marx, Alejandro Muramatsu (Hrsg.)

NIC-Serie Band 10

Winterschule, 25. Februar - 1. März 2002, Rolduc Conference Centre,
Kerkrade, Niederlande

ISBN 3-00-009057-6, Februar 2002, 548 Seiten

**Quantum Simulations of Complex Many-Body Systems:
From Theory to Algorithms - Poster Presentations**

Johannes Grotendorst, Dominik Marx, Alejandro Muramatsu (Hrsg.)

NIC-Serie Band 11

Winterschule, 25. Februar - 1. März 2002, Rolduc Conference Centre,
Kerkrade, Niederlande

ISBN 3-00-009058-4, Februar 2002, 88 Seiten

**Strongly Disordered Quantum Spin Systems in Low Dimensions:
Numerical Study of Spin Chains, Spin Ladders and
Two-Dimensional Systems**

Yu-cheng Lin

NIC-Serie Band 12

ISBN 3-00-009056-8, Mai 2002, 131 Seiten

**Multiparadigm Programming with Object-Oriented Languages -
Proceedings**

Jörg Striegnitz, Kei Davis, Yannis Smaragdakis (Hrsg.)

Workshop MPOOL 2002, 11 Juni 2002, Malaga

NIC-Serie Band 13

ISBN 3-00-009099-1, Juni 2002, 133 Seiten

**Quantum Simulations of Complex Many-Body Systems:
From Theory to Algorithms - Audio-Visual Lecture Notes**

Johannes Grotendorst, Dominik Marx, Alejandro Muramatsu (Hrsg.)

NIC-Serie Band 14

Winterschule, 25. Februar - 1. März 2002, Rolduc Conference Centre,
Kerkrade, Niederlande

ISBN 3-00-010000-8, November 2002, DVD

Numerical Methods for Limit and Shakedown Analysis

Manfred Staat, Michael Heitzer (Hrsg.)

NIC-Serie Band 15

ISBN 3-00-010001-6, Februar 2003, 282 Seiten

**Design and Evaluation of a Bandwidth Broker that Provides
Network Quality of Service for Grid Applications**

Volker Sander

NIC-Serie Band 16

ISBN 3-00-010002-4, Februar 2003, 185 Seiten

**Automatic Performance Analysis on Parallel Computers with
SMP Nodes**

Felix Wolf

NIC-Serie Band 17

ISBN 3-00-010003-2, Februar 2003, 143 Seiten

Alle Bände stehen online zur Verfügung unter <http://www.fz-juelich.de/nic-series/>.